Defence Research and Development Canada
Recherche et développement pour la défense Canada

DEFENCE R&D DÉFENSE

Proactive QoS Routing in Ad-Hoc Networks

Ying Ge, Louise Lamont and Thomas Kunz

The work described in this document was sponsored by the Department of National Defence under Work Unit 5CN.

Defence R&D Canada – Ottawa

TECHNICAL REPORT
DRDC Ottawa TR 2003-075

Communications Research Centre

CRC-RP-2003-005
June 2003

Canada

Proactive QoS Routing in Ad-Hoc Networks

Ying Ge, Louise Lamont
Communications Research Centre

Thomas Kunz
Carleton University

The work described in this document was sponsored by the Department of National Defence under Work Unit 5CN

Defence R&D Canada – Ottawa

TECHNICAL REPORT

DRDC Ottawa TR 2003-075

Communications Research Centre

CRC-RP-2003-005

JUNE 2003

Abstract

Quality-of-service (QoS) routing in mobile Ad-Hoc networks is challenging because the network topology may change constantly and the available state information for routing is inherently imprecise. In this report, the authors have developed different QoS versions of the OLSR (Optimized Link State Routing) protocol, which is a "pro-active" Ad-Hoc routing protocol. They have introduced algorithms that allow OLSR to find the maximum bandwidth path and have shown that these algorithms do improve OLSR in terms of bandwidth. They have also analyzed the performance of the QoS routing protocols in OPNET, observed the results obtained, and the consequences. The simulation results show that the QoS versions of the OLSR routing protocol do improve the available bandwidth of the routes computed, but the added cost – the additional overhead also has a negative impact on the network in End-to-End Delay and Packet Delivery Ratio, especially in the high speed movement scenarios. The authors believe that proactive QoS routing is still worth while studying. Emphasis on future studies should be on how to reduce the overhead of QoS pro-active routing protocols.

Résumé

Le routage de la qualité de service (QS) dans un réseau ad hoc est difficile, car la topologie du réseau peut changer à tout moment et les renseignements disponibles sur l'état du réseau sont imprécis par nature. Pour le présent document, les auteurs ont conçu des versions de la QS pour le protocole OLSR (Optimized Link State Routing), un protocole de routage ad hoc « proactif ». Ils ont introduit des méthodes heuristiques qui permettent au protocole OLSR de trouver la largeur de bande maximale et démontrent par la simulation et la preuve que ces méthodes de recherche améliorent effectivement le protocole OLSR relativement à la QS de la largeur de bande. Ils ont également analysé le rendement des protocoles de routage de la QS dans OPNET, les résultats obtenus et les conséquences. Les résultats de la simulation établissent que les versions de la QS du protocole de routage OLSR augmentent véritablement la largeur de bande disponible des chemins prédéterminés, mais le nombre élevé de calculs et la surcharge du réseau ont également des conséquences néfastes sur le délai de bout en bout et le rapport de remise des paquets, notamment lors de mouvements à grande vitesse. Les auteurs croient néanmoins que l'utilisation du routage proactif de la QS en vaut vraiment la peine. Des études ultérieures seront concentrées sur la réduction de la surcharge causée par le routage ad hoc proactif de la QS.

This page intentionally left blank.

Executive summary

The design of routing protocol to support Quality of Service (QoS) in MANETs (Mobile Ad-Hoc Networks) is a challenge. To support QoS, the link performance characteristics such as delay, bandwidth, jitter, cost, loss rate and error rate must be available and manageable. However, getting and managing the link state information is not trivial in a MANET. The quality of a wireless link changes with the surroundings. The changing environment, the bandwidth and battery limitation, and the mobility of hosts add to the complexity. Further more, it is complex to evaluate the QoS routing performance. Compared to traditional best-effort routing, QoS routing has two additional cost factors – "computational cost" and "protocol overhead". "Computational cost" comes from the more frequent path selection computations, as besides maintaining the source-destination connection, computations are also needed to satisfy the QoS request. Additional "protocol overhead" comes from the need to distribute the updated link state information. There is a trade-off between the QoS performance that the QoS routing protocol achieves and the additional cost it introduces.

In on-demand QoS protocols, a route is found based on specific QoS requirements. Because of the unpredictable nature of Ad-Hoc networks and the requirement for quick reaction to QoS routing demands, it would seem that a proactive protocol is more suitable. When a request arrives, the control layer can easily check if the pre-computed optimal route can satisfy such a request. Thus, waste of network resources attempting to discover routes is avoided. But on the other hand, proactive QoS Ad-Hoc routing may introduce additional overhead, which may affect the performance of the routing protocol. There have been very few studies of the additional overhead a proactive QoS routing protocol introduces and how the routing performance is affected.

This document presents the results of our investigation of these issues. The authors introduce a straightforward method to calculate the available bandwidth over the wireless links. They propose three proactive QoS routing algorithms that enhance the Optimized Link State Routing (OLSR) protocol in terms of bandwidth. They show that the algorithms guarantee the optimal bandwidth path in the static network case. They also analyze the OPNET simulation result of one of the algorithms to determine the performance of the QoS routing algorithms and the additional overhead incurred to obtain such gains. The simulation results show that the QoS versions of the OLSR routing protocol do improve the available bandwidth of the routes computed, but the added cost – the additional overhead, has a negative impact on the network End-to-End Delay and Packet Delivery Ratio. This is especially in the case of high-speed movement scenarios. Nevertheless, the authors believe that it is worthwhile to use proactive routing to support QoS in MANETs.

Louise Lamont, Thomas Kunz, Ying Ge. 2003. Proactive QoS Routing in Ad-Hoc Netorks. DRDC Ottawa TR 2003-075 CRC-RP-2003-005 Defence R&D Canada – Ottawa.

Sommaire

Le routage de la qualité de service (QS) dans un réseau mobile ad hoc (MANET) est difficile. Pour soutenir la QS, il faut que les renseignements sur l'état des liens (comme le délai, la largeur de bande, l'instabilité, le coût, le taux de perte et le taux d'erreur) soient disponibles et utiles. L'acquisition et le traitement de renseignements sur l'état des liens dans un MANET ne sont toutefois pas choses simples, car la qualité d'un lien sans fil change selon les conditions environnantes. De plus, la limitation des ressources et la mobilité des hôtes compliquent davantage la situation. Outre ces problèmes, il est difficile d'évaluer le rendement du routage de la QS. Comparativement au routage traditionnel, le routage de la QS compte deux facteurs de coûts supplémentaires : les « coûts de calcul » et la « surcharge du protocole ». Les « coûts de calcul » découlent du calcul plus fréquent du choix des chemins, par comparaison au maintien de la connexion entre la source et la destination; les calculs sont également essentiels pour répondre aux demandes de la QS. La « surcharge du protocole » s'explique par la nécessité de diffuser les renseignements à jour sur l'état des liens. Il existe un compromis entre le rendement de la QS produit par le routage de la QS et ces facteurs de coûts supplémentaires.

Contrairement au routage ad hoc traditionnel pour lequel les protocoles proactif et sur demande sont proposés, le routage ad hoc de la QS s'effectue en majorité par le routage de la QS sur demande. Dans ces protocoles, un chemin est défini selon des exigences prédéterminées de la QS. Cependant, en raison de la nature imprévisible des réseaux ad hoc et de la nécessité d'une réaction rapide aux demandes de routage de la QS, il semble que la solution du protocole proactif convienne davantage. Dès la réception d'une demande, le niveau automatisation et commandes peut facilement vérifier si le chemin optimal prédéfini satisfait à la demande. Par conséquent, il est possible d'éviter la perte de ressources du réseau consacrées à la découverte de chemins impossibles. En revanche, le routage ad hoc proactif de la QS peut engendrer une surcharge et diminuer le rendement du protocole de routage. Il existe toutefois peu d'études sur l'importance et les conséquences de la surcharge causée par un protocole de routage proactif de la QS.

Le présent document se penche sur ces questions. Les auteurs présentent une méthode simple visant à calculer la largeur de bande disponible pour des liens sans fil. Ils proposent trois algorithmes de routage proactifs de la QS qui augmentent la largeur de bande du protocole OLSR (Optimized Link State Routing). Ils démontrent que les algorithmes garantissent une largeur de bande optimale dans le cas des réseaux statiques. Les auteurs ont analysé les résultats de la simulation OPNET de l'un des algorithmes de routage de la QS et de la surcharge engendrée pour obtenir de tels résultats. Les résultats de la simulation établissent que les versions de la QS du protocole de routage OLSR augmentent véritablement la largeur de bande disponible des chemins prédéterminés, mais le nombre élevé de calculs et la surcharge du réseau ont également des conséquences néfastes sur le délai de bout en bout et le rapport de remise des paquets, notamment lors de mouvements à grande vitesse. Les auteurs croient néanmoins que l'utilisation du routage proactif de la QS en vaut vraiment la peine.

Louise Lamont, Thomas Kunz, Ying Ge. 2003. Proactive QoS Routing in Ad-Hoc Networks. DRDC Ottawa TR 2003-075 CRC-RP-2003-005 R&D pour la défense Canada – Ottawa.

Table of contents

List of figures

List of tables

Acknowledgements

The authors would like to thank the Mobile Networking Group of the Communications Research Centre, for their warm-hearted help and innovative suggestions. Also, we would like to thank the Naval Research Laboratory, in Washington, for providing the OPNET OLSR model and for suggestions on the installation.

This work has been funded by the Defence R& D Canada.

This page intentionally left blank.

I. Introduction

A mobile Ad-Hoc network (MANET [11]) is a dynamic multi-hop wireless network that is established by a group of mobile nodes on a shared wireless channel. In an ad-hoc network, all communication is done over a wireless media, without the use of wired base stations. QoS routing in a mobile Ad-Hoc Network is challenging. To support QoS routing, the link state performance measures such as delay, bandwidth, jitter, loss rate and error rate should be available and manageable. Getting and managing such link state information in a MANET is not trivial because the quality of a wireless link changes quite frequently due to the mobility and variations in the surroundings. In addition, it is complex to evaluate the QoS routing performance. Compared to the traditional best-effort routing, QoS routing has two additional overheads – "computational cost" and "protocol overhead" [2]. "Computational cost" comes from the more frequent path selection computations, since besides maintaining the source-destination connection, additional computations are needed to determine paths that satisfy the QoS demands. The additional "protocol overhead" comes from the need to distribute the frequently updated link state information. There is a trade-off between the performance the QoS routing protocol achieves and the additional cost it introduces.

In the on-demand QoS routing protocols such as [3] and [17], a route is found based on specific QoS requirements. However, the unpredictable nature of Ad-Hoc networks and the requirement for quick reaction to QoS demands makes the idea of a proactive protocol more suitable. When a request arrives, the control layer can easily check if the pre-computed optimal route can satisfy such a request. Thus, waste of network resources when attempting to discover routes is avoided. However, similar to a proactive best-effort routing protocol, a proactive QoS routing may introduce "protocol" overhead. Do these additional overhead have a negative effect on the Ad-Hoc network? If yes, then how much additional overhead does a proactive routing protocol introduce into the network? How does the additional overhead affect the performance of the routing protocol? Can we minimize the cost to achieve better performance? Or should we just give up on proactive QoS routing? In this report, the authors investigate the answers to these questions. They study proactive QoS routing, modify a best-effort proactive routing protocol OLSR [7] for bandwidth QoS purposes, and evaluate the performance of the proactive QoS routing algorithms that are proposed.

The rest of the report is organized as follows: Section II briefly introduces QoS (quality-of-service). Section III proposes three algorithms that enhance OLSR in bandwidth aspect. Section IV tests the algorithms in a statistic network case, and proves the optimality of two of the algorithms in that statistic network model; Section V describes the implementation of QoS OLSR model in OPNET; Section VI compares the performance of one of the QoS OLSR algorithms with different parameters and the original OLSR protocol in the dense network case (network containing 50 nodes), and analyzes the overhead introduced and the achievements gained for the QoS routing; Section VII concludes the report and suggests for future work.

II. QoS and QoS Routing

2.1 What is QoS

Quality-of-service (QoS) is the quantitatively defined performance agreement between the service provider and user applications based on the connection requirements. The QoS requirements of a connection are described in a set of performance or metrics and their constrains such as bandwidth (available bandwidth) constraint, delay constraint, jitter constraint, loss ratio constraint, and so on. These QoS requirements, also called QoS metrics, can be "concave" or "additive".

[3] gives the definition of "concave" and "additive" QoS metrics: Let m(i,j) be a QoS metric for link (i,j). For a path P=(s,i,j,…,l,t), metric m is concave if m(P) = min{m(s,i), m(i,j),…,m(l,t)}. Metric m is additive if m(P) = m(s,i)+m(i,j)+…+m(l,t).

Based on the above definition, the bandwidth request is "concave" – the (available) bandwidth of a connection is the minimum of the (available) link bandwidth over the links along the path – which is also called the bottleneck bandwidth of the path. Delay and jitter metrics are "additive". The loss ratio constraint, however, is more complex: the loss ratio of the path (link_a, link_b,…link_n) = 1- (1- loss ratio of link_a) x (1- loss ratio of link_b) x…x (1-loss ratio of link_n).

The QoS condition of a network reflects the network's ability to provide the specified service between communication pairs. Because of the rising popularity of multimedia applications and real-time services, which require strict bandwidth/delay constraints, together with the potential commercial usage of Ad-Hoc networks, QoS support in the MANET has become a topic of interest in the wireless area.

2.2 QoS Routing in Ad-Hoc Networks

Many components should work together to support QoS in Ad-Hoc networks [19]: a QoS model specifies which kinds of services to be supported in the network; a QoS routing scheme searches a path with satisfactory resources defined by the QoS model; a QoS MAC protocol [9], [12], [18] solves the problems of medium contention; a QoS signaling protocol performs the resource reservation along the path computed by the QoS routing protocols. Among all these components, QoS routing is a key issue.

The goals of QoS routing are 1) selecting one or more network paths that have sufficient resources to meet the QoS requirement of connections, 2) provide resource information about the path for the admission control (call acceptance) mechanism, and 3) achieving global efficiency in resource utilization.

QoS routing in Ad-Hoc network is a challenging problem. The challenge is to implement QoS functionality with limited resources in a dynamic environment.

First, to support QoS, the link state information such as the delay, bandwidth, jitter, cost, loss ratio and error ratio must be available and manageable. However, getting and managing the link state information in a MANET is by all means not trivial because the quality of a wireless link changes with the surrounding circumstance. The larger the size of the network, the more difficult it is to gather up-to-date information. Second, resource limitations (operation buttery limitation, bandwidth limitation, etc) and the mobility of hosts complicate the problem. Third, if the QoS request includes two independent path constraints, path searching becomes NP-complete [20]. Besides the above difficulties in QoS route computation, it is also complex to evaluate the QoS routing performance – network topology and traffic characteristics effect the performance of QoS routing. QoS routing may be more effective in networks with uneven traffic load; different network topologies may also effect the performance of routing

algorithms [2]. Even if the QoS routing protocol successfully enhances the network performance, it is worthwhile to question if it is worth the cost. Compared to traditional best-effort routing, QoS routing has two added cost factors – "computational cost" and "protocol overhead" [2]. "Computational cost" comes from the more frequent path selection computations. Besides maintaining the source-destination connection, computations are also needed to satisfy the QoS request. "Protocol overhead" comes from the need to distribute the updated link state information. The trade-off between the performance the routing protocol achieves and the additional cost it introduces should be carefully observed and well understood.

2.3 Related Work

From the literature, not much discussion is made on the overhead the QoS routing algorithms introduces and its impact on the performance of the routing protocols. Among the known on-demand QoS routing protocols, [3] shows the performance of the "ticket-based probing" algorithm in a delay-constrained environment, calculating what percentage of the routes that the algorithm finds meet the delay request. But it fails to analyze other aspects of the routing algorithm, such as control overhead, packet delivery ratio etc. [17] tests the CEDAR algorithm using bandwidth as the QoS parameter, giving the performance evaluation on message complexity for route computation, packet delivery ratio, bandwidth optimal ratio (difference between the bandwidth over the paths the routing algorithm computed and the largest available bandwidth paths in the network). However, [17] does not do experiment with node movement. Nor does it run the simulation in a real shared-channel environment, and the impact of channel interference and packet collision are not considered.

Many proposed proactive QoS routing algorithms such as [5] and [16] just present a basic idea, without performance evaluation.

In this report, the authors demonstrate, by using simulation, not only the QoS performance of the original OLSR protocol and the QoS OLSR versions, but also their overhead as well.

III. OLSR and QoS OLSR

This section briefly describes the OLSR algorithm, and proposes three heuristics that enhance OLSR when considering bandwidth as the QoS constraint.

3.1 Description of OLSR

In [7], the IETF MANET Working Group introduces the Optimized Link State Routing (OLSR) protocol for mobile Ad-Hoc networks. The protocol is an optimization of the pure link state algorithm. The key concept used in the protocol is that of Multi-Point Relays (MPRs) introduced in [6] and [15]. MPRs are selected nodes that forward broadcast messages during the flooding process. This technique substantially reduces the message overhead as compared to a pure flooding mechanism where every node retransmits messages throughout the network. By doing so, the "contents" of the control messages flooded in the network are also minimized. So contrary to the classic link state algorithm, instead of all links, only small subsets of links are declared.

OLSR operates as a table-driven and pro-active protocol. The node n, which is selected as a multipoint relay by its neighbors, periodically announces the information about who has selected it as an MPR. Such a message is received and processed by all the neighbors of n, but only the neighbors who are in n's MPR set retransmit it. Using this mechanism, all nodes are informed of a subset of links -- links between the MPR and MPR selectors in the network. For route calculation, each node calculates its routing table using a "Shortest Hop Path" algorithm based on the partial network topology it has learned. The algorithm finds the minimum hop path from the source node to all the destinations. In addition to re-transmitting topology control messages, the MPRs are used as a backbone network to form the route from a given node to any destination in the network.

MPR selection is the key point in OLSR. The MPR set is selected such that it covers all nodes that are two hops away. This means that the union of the neighbor sets of the MPRs contains the entire 2-hop neighbor set of a node. Each node selects its MPRs independently. The smaller the MPR set, the less overhead the protocol introduces. The proposed heuristic in [7] is as follows:

1. start with an empty MPR set

2. for each node y in the 1-hop neighbor set N, calculate D(y) – the degree (the number of neighbors) of y

3. select as MPRs those node in N which provide the "only path" to some nodes in the 2-hop neighbor set N2

4. while there exist nodes in N2 which are not covered

   { Select as an MPR a 1-hop neighbor, which reaches the maximum number of uncovered nodes in N2. If there is a tie, the one with higher degree is chosen.}

5. As an optimization, process each node y in MPR. If MPR\{y} still covers all nodes in N2, y should be removed from the MPR set.

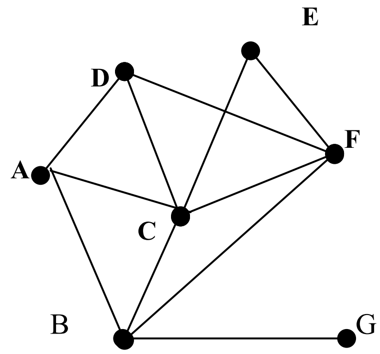


Figure 1: *Network Example for MPR Selection*

An example of how this algorithm works is shown below based on the network depicted in Figure 1.

Table 1. *MPR Selection in the Original OLSR*

Nodes	1 hop Neighbors	2 hop Neighbors	MPR(s)
B	A, C, F, G	D, E	C

From the perspective of node B, both C and F cover all of node B's 2-hop neighbors. However, C is selected as B's MPR as it has 5 neighbors while F only has 4 (C's degree is higher than F).

3.2 Integrating OLSR and QoS Routing

3.2.1 Limitations of OLSR in QoS Routing

OLSR is a routing protocol for best-effort traffic, with emphasis on how to reduce the overhead, and at the same time, provide a minimum hop route. So in its MPR selection, the node selects the neighbor that covers the most unreached 2-hop neighbors as MPR. This strategy limits the number of MPRs in the network to ensure that the overhead is as low as possible. However, for QoS routing, using such an MPR selection mechanism, the "good quality" links may be "hidden" from other nodes in the network. As an example, consider the network topology in Section 3.1 again (see Figure 2.) The numbers along the lines indicate the available bandwidth over the links. As explained in Section 3.1, in the OLSR MPR selection algorithm, node B will select C as its MPR. So all the other nodes only know that B can be reached via C. Obviously, when D is building its routing table, for destination B, it will select the route D-C-B, whose bottleneck bandwidth is 3, the worst among all the possible routes.

When "bandwidth" is considered to be the QoS constraint, in building the routing tables, nodes can no longer use the "Shortest Hop Path" algorithm proposed in [7], as the path with the minimum hops may not be the path with best bandwidth. To overcome these limitations, revisions are proposed to two features: MPR selection and routing table computation. These are described separately in the following two subsections.

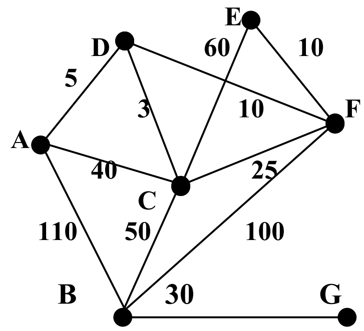


Figure 2: *Bandwidth-QoS Network Example for MPR Selection*

3.2.2 Changing the MPR Selection Criteria

The decision on how each node selects its MPRs is essential to determining the optimal bandwidth route in the network. In MPR selection, a "good bandwidth" link should not be omitted. In other words, as many nodes as possible that have high bandwidth links connecting to the MPR selector must be included in the MPR sets. Three revised MPR selection algorithms based on this idea are presented.

3.2.2.1 OLSR_R1

In the first algorithm, MPR selection is almost the same as that of the original OLSR described in Section 3.1. However, when there is more than one 1-hop neighbor covering the same number of uncovered 2-hop neighbors, the one with the largest bandwidth link to the current node is selected as MPR:

1. start with an empty MPR set

2. select as MPRs those nodes in N which provide the "only path" to some nodes in 2-hop neighbours N2

3. while there exists nodes in N2 which are not covered

   {select as an MPR a 1-hop neighbour which reaches the maximum number of uncovered nodes in N2. If there is a tie, the one with higher bandwidth is chosen}

4. As an optimization, process each node y in MPR. If MPR\{y} still covers all nodes in N2, y should be removed from the MPR set.

The network in Figure 2 would select MPRs for node B as follows, based on OLSR_R1:

Table 2. *MPR Selection in OLSR_R1*

Nodes	1 hop Neighbors	2 hop Neighbors	MPR(s)
B	A, C, F, G	D, E	F

3.2.2.2 OLSR_R2

The idea behind OLSR_R2 is to select the highest bandwidth neighbors as MPRs:

1. start with an empty MPR set

2. selects as MPRs those nodes in N which provide the “only path” to some nodes in 2-hop neighbours N2

3. while there exists nodes in N2 which are not covered

   {select as an MPR a node that has the highest bandwidth link connected with the current node. If there is a tie, the one that covers more uncovered 2-hop neighbours is selected}

For example, using this algorithm, based on Figure 2, node B’s MPR(s) would be:

Table 3: *MPR Selected in OLSR_R2*

Nodes	1 hop Neighbors	2 hop Neighbors	MPR(s)
B	A, C, F, G	D, E	A, F

Among node B’s neighbors, A, C, and F have a connection to its 2-hop neighbors. Among them, link BA has the largest bandwidth. So A is first selected as B’s MPR, and the 2-hop neighbor D is covered. Similarly, F is selected as MPR next and E is covered, so all 2-hop neighbors are covered and the algorithm terminates.

3.2.2.3 OLSR_R3

The third algorithm selects the MPRs in a way such that all the 2-hop neighbors have the optimal bandwidth path through the MPRs to the current node. Here, optimal bandwidth path means the bottleneck bandwidth path is the largest among all the possible paths.

1. start with an empty MPR set

2. selects as MPRs those nodes in N which provide the “only path” to some nodes in 2-hop neighbours N2

3. while there exists nodes in N2 which are not covered

   {select as an MPR a node so that the current node has the optimal route through the MPR to a 2-hop node}

Look again at node B in Figure 2 as an example. In order to cover D, neighbors A, C, or F need to be chosen as an MPR. Bandwidths available from B to D for three different routes are:

B –110- A –5- D bottleneck bandwidth is 5
B –50- C –3- D bottleneck bandwidth is 3
B –100- F –10- D bottleneck bandwidth is 10

The algorithm chooses the route with the largest bottleneck (in 2 hops). In this case the chosen MPR is F. In the same way, C is chosen as MPR by B to cover E.

Table 4: *MPR Selected in OLSR_R3*

Nodes	1 hop Neighbors	2 hop Neighbors	MPR(s)
B	A, C, F, G	D, E	F, C

The three revised OLSR MPR selection algorithms may improve the chance that a better bandwidth route is found. However, the overhead may also increase compared with the original OLSR algorithm because the number of MPRs in the network may be increased. This is especially the case for OLSR_R3, which may select a different MPR for each 2-hop neighbor.
These algorithms are analyzed in the simulations done for the static network model and the mobile Ad-Hoc network model, to determine what kind of improvement can be obtained and what price (in terms of the additional overhead) must be payed for the achievement.

3.2.3 Routing Table Calculation

Besides the MPR selection method, a node also needs to change the “Shortest Hop Path” algorithm in its routing table computation to reflect the bandwidth as the QoS metric. Here, the extension of the Bellman-Ford shortest path algorithm presented by [4] is used to calculate the “shortest-widest” path – the best bandwidth path from a source to any reachable destination with minimum hop count.

In this section, the algorithms that enhance the OLSR protocol in QoS aspect are proposed. In the following sections, the advantage and disadvantage of these algorithms are analyzed through simulations.

IV. QoS OLSR Evaluation in Static Networks

This section gives the simulation results based on the static network case and proves that two of our algorithms proposed in Section 3 are indeed optimal, i.e., guarantee that the bandwidth-optimal path is found.

4.1 Static Network Simulation Result

In this sub-section, simulation of the MPR selection algorithms is discribed and the results compared. It is assumed that the Ad-Hoc network topology is stable at one moment so that the QoS routing problem can be studied on that stable graph. Actually, there are various circumstances where networks are rather stable: A wireless network consisting of desktops, laptops and printers for home business may keep its original topology for a long time until someone moves one of the laptops to another room, for example. In the next chapter, the algorithms are tested in a simulated mobile network environment to see what impact node movements have on link-state updating and the network performance.

Assuming the devices in an Ad-Hoc network are configured with the same wireless card, then all the nodes in the network have the same maximum bandwidth. It is only interesting that how much of the remaining bandwidth is available for new traffic. Many papers such as [10] discuss how to compute bandwidth in Ad-Hoc networks. Here, a rather simple and straightforward approach similar to [1] is used: measuring how much time a node monitors an idle channel and thus is available to transmit new messages over a link (node's idle time). MAC protocols such as IEEE 802.11 are based on a carrier-sense capability of each node. This capability is exploited to determine, locally at each node, for what percentage of time the medium has been busy in the recent past. A busy medium may indicate that a neighbor is transmitting data over the shared wireless channel. However, it may also indicate that nodes even further away, but still within interference range, are using the media. A node can only successfully transmit during times when neither its immediate neighbors nor other nodes in its interference range are transmitting. This characterization of the available bandwidth hasS lower overhead than proposals where nodes communicate with their immediate neighbors to exchange information about their committed bandwidth, ignoring nodes further away. The "available bandwidth" over a link connecting nodes A and B is proportional to the minimum of A's idle time and B's idle time, since both nodes have to be available for a successful transmission. Since the number of nodes and the traffic between them in each node's interference range is different, the idle times of two adjacent nodes may well be substantially different. However, due to the shared nature of the wireless medium, it is always the case that the link bandwidth between two adjacent nodes A and B is always equal to or better than the bandwidth over any 2-hop connection between A and B (i.e., via some intermediate node C), as will be explained in more detail in Section 5.2. Depending on the underlying MAC protocol, a node may not be able to use the whole idle time. In IEEE 802.11 networks, for example, a node will wait for a random backoff time after it detects that the link is idle. However, as such backoff times are deliberately kept short, they are neglected in the remainder of this report. Because of the unstable nature of Ad-Hoc networks, it is also important to decide how the idle time, which reflects the network traffic condition, should be maintained and updated. This issue will be addressed in the next section. In this section, the "network snapshots" are studied to evaluate the route selection heuristics in OLSR.

Using C++, a simple platform is written to simulate the QoS OLSR algorithms. The platform randomly generates nodes and defines their positions and the available link bandwidths. Then,

based on the nodes position and bandwidth information, routes are computed based on the algorithms proposed and the original OLSR separately. The networks generated by the platform are fixed graphs, which represent snapshots of the Ad-Hoc network state. The following are the simulation details:

4.1.1 Network Scenario

- Network area: 1000M x 1000M
- Number of nodes: 100
- Transmission range: 100M, 200M, and 300M
- Bandwidth: Based on the analysis in this section, the available bandwidth is computed as follows: Each node is randomly assigned an "idle time" ranging from 0 to 1. The available link bandwidth between two nodes is equal to the minimum of their idle time × maximum bandwidth. Here, it is considered that in the Ad-Hoc network, each link has the same maximum bandwidth, 2 Mbps. For example, if node a's idle time is 0.5 and node b's idle time is 0.3, then the available bandwidth over link ab is: 0.3 × 2Mbps = 600 kbps. These randomly generated "idle times" reflect the traffic condition in the network snapshot because the consumed bandwidth over each link reflects the traffic flows over that link.

4.1.2 Simulation Objective

A total of 5 algorithms are implemented and applied to the randomly generated network snapshots:
1) OLSR (Section 3.1) with "shortest hops path" route computation algorithm
2) OLSR_R1 (Section 3.2.2.1)
3) OLSR_R2 (Section 3.2.2.2)
4) OLSR_R3 (Section 3.2.2.3)
(The above 2-4 are all using the "Extended BF" algorithm for route computation)
5) Pure link state algorithm: each node floods its link state information into the entire network. Then, the best bandwidth routes are computed with the "Extended BF" algorithm. By doing this, the path with maximum bottleneck bandwidth is guaranteed to be found.

Routes found by algorithms 1) through 4) are compared with the route found by algorithm 5), using the simulation model and metrics discussed below.

4.1.3 Simulation Model

For each transmission range (100m, 200m, 300m), 100 network snapshots are generated. For each connected pair in the network, the 5 algorithms mentioned in Section 5.1.2 are run to find a route between each pair of nodes in the network. Results obtained show how often the route found by the first 4 algorithms (original OLSR, OLSR_R1, OLSR_R2, and OLSR_R3) has lower bandwidth than the route found by a pure link state algorithm. If the optimal path can not be found using the first 4 algorithms, how sub-optimal the result

is shown. Also, the overhead of these 5 algorithms is characterized and compared.

4.1.4 Simulation Results

Results are given in two categories: performance and cost. To further analyze the results, information about specific network characteristics is also collected.

4.1.4.1 Performance

Performance is characterized by "Error Rate" and “Average Difference”:

- “Error Rate” represents the percentage of times the standard OLSR, OLSR_R1, OLSR_R2, and OLSR_R3 algorithms do not find the optimal bandwidth path. In other words, Error Rate = total number of bad routes in 100 snapshots computed by OLSR / total number of optimum routes in 100 snapshots.

- “Average Difference” is the average of the difference between the optimal bandwidth and current bandwidth found in routing algorithms in percentage: result = average of (bandwidth on optimal path-bandwidth on route computed)/bandwidth on optimal path, when the optimum routes are not found. The larger the value is, the worse the result.

4.1.4.2 Cost

The cost of the protocol is measured by “overhead” and “MPR percentage”:

- “Overhead”: How many control messages (messages originated by the nodes indicating who select it as MPR) are transmitted/re-transmitted in the network. Overhead = the average number of control messages transmitted per snapshot/100 (the number of nodes in network).

- “MPR Number”: Average number of MPRs in the network. The more MPRs in the network, the higher the overhead.

4.1.4.3 Network Characteristics

The average number of 1-hop neighbors and 2-hop neighbors for a node is collected. These values affect the MPR number in the network. On one hand, the more 1-hop neighbors a node has, the less MPRs it may select, because with a high probability a small subset of its 1-hop neighbor can reach a high number of the 2-hop neighbors (assuming high connectivity of the network). On the other hand, the more 2-hop neighbors a node has, the more MPRs may be needed to cover them all.

4.1.4.4 Simulation Results and Analysis

Simulation Results are presented in Table 5 and Table 6.

Table 5: *Network Characteristics*

Transmission range	300M	200M	100M
Number of1-hop neighbors	21	10	2
Number of 2-hop neighbors	33	15	4

Table 6: *Summary of Simulation Results*

Algorithm	Transmission Range	Performance		Cost	
		Error Rate	***Average Difference***	***Overhead***	***MPR Number***
Original OLSR	***300 M***	28%	46%	12	65
	200 M	41%	51%	24	68
	100 M	12%	45%	5	42
OLSR_R1	***300 M***	14%	22%	12	65
	200 M	21%	26%	24	68
	100 M	8%	44%	5	42
OLSR_R2	***300 M***	0%	0%	18	70
	200 M	0%	0%	33	72
	100 M	0%	0%	5.7	45
OLSR_R3	***300 M***	0%	0%	26	71
	200 M	0%	0%	38	73
	100 M	0%	0%	5.7	44
Pure Link State Algorithm	***300 M***	0%	0%	1245	100
	200 M	0%	0%	979	100
	100 M	0%	0%	28	100

- First the results of all 5 algorithms for the same network are considered, using the 300 M transmission range network as example (see Table 6):

  Considering the performance of the 4 OLSR algorithms, it is shown that the original OLSR has the worst performance – it has the highest “Error Rate” and “Average Difference”, which means in the 300 M transmission range network, the original OLSR has the highest probability that it can not find the best bandwidth path. At the same time, the bandwidth difference between the paths it finds and that of the optimal path is also large. Although the OLSR_R1 uses the same MPR selection algorithm as the original OLSR, it achieves a large improvement in performance, which shows lower “Error Rate” and lower “Average Difference”. Such improvement is affected by the “Extended BF” algorithm, which finds the optimal path on the partial network a node learns from the procedure of MPR selector declaration and re-transmission. However, OLSR_R1 does not always find an optimal path, as

its MPR selection algorithm may omit the optimal bandwidth link from the partial network topology the node learned. (See the example of Section 3.2.1). However, OLSR_R2 and OLSR_R3 show very good results – each time, these two algorithms find the optimal bandwidth route. The explanation for this extremely good result is given in Section 4.2.

As mentioned earlier, costs are directly related to the number of MPRs selected by the algorithms. The higher the number of MPRs in the network is, the higher the overhead. This relationship is clearly shown in the "Cost" category. Of the 5 algorithms, in its MPR selection, standard OLSR emphasizes on reducing the number of MPRs in the network to lower the overhead. so it has the lowest MPR number and overhead compared with OLSR_R2, OLSR_R3 and Pure Link State Algorithm. (OLSR_R1 has almost the same MPR selection mechanism as that of standard OLSR, and these two algorithms therefore have comparable overheads.) Also, as predicted in Section 3.2.2, OLSR_R2 and OLSR_R3 select more MPRs, thus produce higher overhead than standard OLSR. Compared with OLSR_R2, OLSR_R3's overhead is even higher, which is also consistent with our prediction. For Pure Link State algorithm, it obviously has the highest overhead, with each node acting as MPR, re-transmitting the messages it receives.

The result of all 5 algorithms in networks with a transmission range of 200 M and 100 M network have similar characteristic as the 300 M transmission range case.

- The performance of the individual algorithms is also explored:

  Standard OLSR: At first glance, it may seem strange that a network with a node transmission range of 200 M has the highest overhead. Intuitively, the denser the network is, the higher the overhead: for the same number of nodes and area size, the network contains more edges (in other words, a node will have more neighbors) if the transmission range of a node is higher (see Table 5). However, the result can be explained as follows: in general, the more MPRs are selected, the higher the overhead. In a higher density network (such as for a node transmission range of 300 M), node connectivity is also high, so a node may need fewer MPRs to cover its 2-hop neighbors. On the contrary, in lower density network (such as for a node transmission range of 100 M), because of the lower connectivity, a node may have fewer 2-hop neighbors; therefore, it also needs fewer MPRs. However, the transmission range of 200 M falls within these two extremes, so it may well result in the largest number of MPRs to produce the highest overhead. This situation is not found in the Pure Link State Algorithm, where a node's entire neighbor set is its MPR set. So the denser the network is, the more neighbors/MPRs a node has, resulting in a higher overhead.

  Also, one may expect that the denser the network is, the worse the performance should be. With higher connectivity, there are more possible routes from a given source to a destination, and the probability that OLSR chooses a non-optimal route is higher. This tendency can be seen when comparing the performance of 300 M and 100 M transmission range networks. But again the 200 M transmission range network is the exception, having the highest "Error Rate". Considering a node in an optimal bandwidth route, its next hop node on the path is its 1-hop neighbor, and the hop after next is its 2-hop neighbor (proof is given in Section 5.2). In other words, an

optimal bandwidth path is composed of segments "node->1-hop neighbor -> 2-hop neighbor". The route computed by OLSR has that feature as well. For 100 M transmission range, because of its lower connectivity, the node has less 1-hop neigbhors and 2-hop neighbors. As a result, in this network, there are fewer segments of "node->1-hop neighbor -> 2-hop neighbor", resulting in a lower propability that OLSR chooses the wrong path. For the dense network (300 M transmission range), a node has many more 1-hop and 2-hop neighbors, resulting in many segments of "node->1-hop neighbor -> 2-hop neighbor". The selected MPRs will cover many of the 2-hop neighbours more than once, again resulting in a lower propability for OLSR ignoring the segments belonging to the optimal path. As shown by the difference between OLSR and OLSR_R1, a simple change in how to calculate the paths, based on the same MPR set, can yield significant performance improvements. Again, the 200 M transmission range case falls between these two extremes, resulting in the worst performance.

OLSR_R1: the result shows the same trends as that of the original OLSR. Also, when comparing the performance of the original OLSR and OLSR_R1, it shows that OLSR_R1 achieves larger improvements over the original OLSR in higher density network. That is because for higher density networks, more links are declared to a node. So when computing its routing table, a node has more choices in path selection. The original OLSR uses the Shortest Hops Path Algorithm for route computation, which is unsuitable for bandwidth QoS routing. So the probability that the original OLSR picks up a non-optimal path is higher in denser networks.

OLSR_R2 and OLSR_R3: Regarding performance, they both find the optimal path. Regarding the cost, they also exhibit the phenomenon that a 200 M transmission range network has the highest MPR number/overhead. The reason is the same as the one explained above for standard OLSR.

Pure Link State Algorithm: Comparing the original OLSR with the Pure Link State Algorithm, it is found that the higher the network density, the more obvious the overhead reduction is achieved by the original OLSR. This is consistent with the declaration in [7] that the denser the network is, the more optimization OLSR will achieve, compared to the Link State Algorithm.

4.2 Correctness of the Revised OLSR Algorithm

From the simulation results, it is found that under the current simulation model, both OLSR_R2 and OLSR_R3 always find the optimal path. Can these two algorithms guarantee the optimal result? This is indeed the case. Following is the proof:

Theorem 1: OLSR_R2 finds the optimal bandwidth path.

LEMMA 1: The intermediate nodes on one of the optimal paths (the path with the highest bottleneck bandwidth) are all selected as MPRs by the previous nodes on the path.

Proof: A node in the route may not be selected as the MPR by the previous node if: 1) the node does not provide connection to that node's 2-hop neighbors and 2) the node does not meet the MPR selection criteria. In the following proof, these two situations are addressed separately.

1) A direct link between two nodes a and b always has same or better available bandwidth than any routes connecting a and b via some intermediate nodes.
   Proof: In Figure 3, there are two paths from a to b: link (ab) and link (a, n_1, n_2, n_3,…n_k, b). Suppose node a, b, n_1, n_2, n_3,…n_k's idle time are I_a, I_{n1}, $I_{n2,}$ $I_{n3,}$ $I_{nk,}$ I_b respectively.

As discussed in Section 4.1, the wireless medium studied here is the shared channel. A node can only successfully transmit during times no nodes in its interference range are transmitting (the channel is idle), and as both the two nodes a and b on the link ab should be available during the transmission, which means that the bandwidth over link ab should be $\min(I_a, I_b)$. And also, it is supposed here that all the nodes in the network are configured with same data rate. So based on the concave nature of the available bandwidth, bandwidth of link (AB) and link ($A, N_1, N_2, N_3, \ldots N_k, B$) are

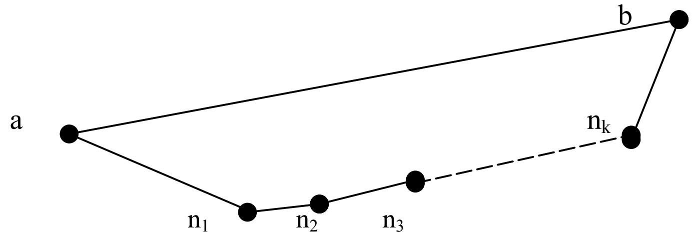


Figure 3: *Two Different Paths Connect Node a and Node b*

- Link (ab): $\min(I_a, I_b)$
- Link(a, n1, n2, n3,...nk, b) : min of bandwidth on links(AN_1, N_1N_2, N_2N_3, ...N_kB) $= \min(I_a, I_{n1}, I_{n2}, I_{n3}, \ldots I_{nk}, I_b)$

It is clear that link (AB) provides the same or better bandwidth path because $\min(I_a, I_b) \geq \min(I_a, I_{n1}, I_{n2}, I_{n3}, \ldots I_{nk}, I_b)$

→ The direct path connecting two nodes has the same or better available bandwidth than the path via any intermediate nodes.

Also, conclusion can be drawn that if a node has no connection to its neighbors' 2-hop neighbors, it is not on the optimal path, as this is the path via the intermediate node (the 1-hop neighbor that connects to another 1-hop neighbor).

2) There is an optimal path from source to destination such that all the intermediate nodes on the path are selected as MPR by their previous nodes on the same path.
Proof: Without loss of generality, it is supposed that in an optimal path, $S, M_1, M_2 \ldots M_k, M_{k+1}, \ldots M_r, D$, there are nodes in the route which are not selected as MPRs by their previous nodes. Also, based on the result of 1), it can be assumed that for each node on the path, its next node on the path is its 1-hop neighbor, and the node two hops away from it is its 2-hop neighbor. For example, M_1 is S's 1-hop neighbor, M_2 is S's 2-hop neighbor. M_{k+1} is M_k's 1-hop neighbor, M_{k+2} is M_k's 2-hop neighbor, etc (see Figure 4).

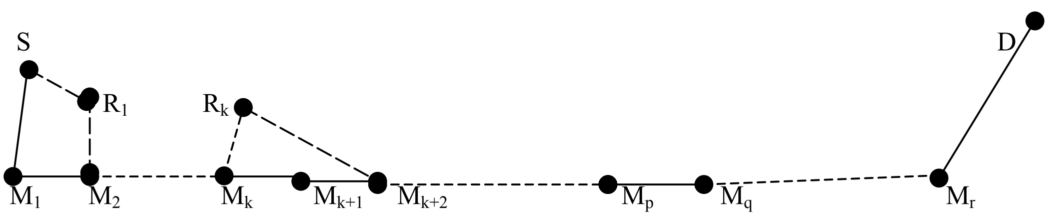


Figure 4: *Route from Source S to Destination D*

a) Suppose that on the optimal route, the first intermediate node M_1 is not selected as MPR by source S. However, M_2 is the 2-hop neighbor of S. Based on the basic idea of MPR selection that all the 2-hop neighbors of a node should be covered by this node's MPR set, S must have another neighbor R_1, which is selected as its MPR, and is connected to M_2. According to the criteria of MPR selection specified in OLSR_R2, S selects R_1 instead of M_1 as its MPR because the link bandwidth of SR_1

is better than the link bandwidth of SM_1, which means Ir_1 (idle time of node R_1) is larger than or equal to Im_1 (idle time of node M_1).

Define bottleneck bandwidth of route R as B(R).

B(S->R_1->M_2->…->M_r->D)

= min(**B(S->R_1->M_2)**, **B(M_2->…->D)**)

= min(min(Is, Ir_1, Im_2), **B(M_2->…->D)**)

B(S->M_1-> M_2->…->D)

= min(min(Is, Im_1, Im_2),**B(M_2->…->D)**)

As $Ir_1 \geq Im_1$, min(Is, Ir_1, Im_2) ≥ min(Is, Im_1, Im_2)

→ **B(S->R_1->M_2->…->M_r->D) ≥ B(S->M_1->…->D).**

Based on our assumption, route S->M_1-> M_2->…->D is optimal path

→ S->R_1-> M_2->…->D is also an optimal path

→ Source's MPR are on the optimal path.

b) Assume that on the optimal route S->M_1->…->M_k->…->D, all the nodes on segment M_1->M_k are selected as MPR by their previous nodes, we now prove that the next hop node of M_k on the optimal route is M_k's MPR.

Suppose that M_{k+1} is not M_k's MPR. Same as above, M_{k+2} is the 2-hop neighbor of M_k, so M_k must has another neighbor R_k, which is the MPR of M_k and has connection to M_{k+2}.

Again, M_k selects R_k instead of M_{k+1} as its MPR because link bandwidth M_kR_k is better than M_kM_{k+1}, which means Ir_k (idle time of node R_k) is better than Im_{k+1} (idle time of node M_{k+1}).

B(S->...M_k->M_{k+1}->M_{k+2}->…M_r->D)

= min(**B(S->M_k)**, min(Im_k, Im_{k+1}, Im_{k+2}), **B(M_{k+2}->D)**)

B(S->…M_k->R_k->M_{k+2}…->D)

= min(**B(S->M_k),** min(Im_k, Ir_k, Im_{k+2}), **B(M_{k+2}->D)**)

≥ **B(S->...M_k->M_{k+1}->M_{k+2}->…M_r->D)**

As S->…->M_k->M_{K+1}->M_{k+2}->…->D is optimal route

→ S->…->M_k->R_k->M_{k+2}…->D is also optimal route.

→ In an optimal route, the (k+1)th intermediate node is the MPR of the (k)th intermediate node.

Based on a) and b), all the intermediate nodes of an optimal path are the MPRs of the previous nodes.

LEMMA 2: A node can correctly compute the optimal path for the whole network topology.

Proof:

1) As discussed in Section 3.2.3, using "Extended BF Algorithm", a node can compute the optimal path on the known partial network topology
2) In OLSR, each node knows the links between MPRs and their selectors in the network. Based on LEMMA 1, there is an optimal path such that all the intermediate nodes on it are the MPR of the previous node on the same path. So the optimal path for the whole network topology is included in the partial topology the node knows.

→ The node can correctly compute the optimal path for the whole network topology.

Theorem 2: OLSR_R3 finds the optimal bandwidth path.

The proof is similar to that of Theorem 1.

In this section, simulations are done for the QoS OLSR algorithms on network snapshots using a simple simulation platform that developed by the authors. The achievements of the QoS OLSR algorithms made in bandwidth aspect are shown. Actually, in the static network case, two of the algorithms are guaranteed to find the optimal bandwidth path. However, these

achievements are made without the consideration of the impact of data traffic and node movement. Will these algorithms still outperform the original OLSR in a real Ad-Hoc Network environment? In the next two sections, analysis will be done on the performance of one of the algorithms simulated in OPNET, which provide a “real” mobile Ad-Hoc network environment.

V. OLSR and QoS OLSR Model in OPNET

Section IV compares the performance of original OLSR protocol and the QoS OLSR versions in the static network case using a simple platform written by us. In this and the following section, the OLSR algorithms are studied using OPNET to determine the algorithms' performance with node movements and data flows.

5.1 Introduction to OPNET

At the time when the project began, only the OPNET OLSR model is available. So OPNET is chosen to implement the QoS functionality of OLSR.

Originally developed at MIT, OPNET [13] is a network simulator allowing researchers to design and study communication networks, devices, protocols, and applications. An OPNET simulation package includes three main graphic editors – network editor, node editor, and process editor. The network editor manages network topologies; the node editor controls network devices' performance; the process editor implements protocols, resources, applications, algorithms, and queuing policies. These three editors work together to provide various simulation environments.

In OPNET, the Wireless LAN protocol is based on the IEEE 802.11 carrier sense multiple access and collision avoidance (CSMA/CA) distributed coordination function (DCF) access scheme. The unicast data packets are transmitted with the RTS/CTS frame exchange to reserve media, and the "Data and Acknowledgement" frame exchange to ensure the transmission reliability. The broadcast data packets, however, can be transmitted after sensing an idle channel, but may suffer from the collision by the hidden-terminal problem. In the simulation, modifications are done to the OPNET Wireless LAN model to calculate the available bandwidth, This is discussed in section (Section 5.2).

5.2 OLSR Model in OPNET

5.2.1 The Original OPNET OLSR Model

The original OLSR model for OPNET was developed by the Naval Research Laboratory (NRL) of the United States Department of Defense. Figure 5 shows an OLSR node in OPNET Node Editor.

In the above OLSR node model, except for the "olsr" process model and the "udp_gen" process model (see the box in upper part of Figure 5), all the other process models are the standard process models of OPNET.

- "olsr" process model

The "olsr" process model implements the OLSR routing protocol discussed in Section 3.1. The following figure (Figure 6) shows the OLSR implementation in the OPNET Process Model.

After initialization and sending an empty Hello message to begin the process, the OLSR routing protocol continuously goes to "itimer" state to decide if it is time to send a Hello message or a TC message. If yes, the message is sent and olsr returns to "idle" state. When a packet (Hello message or TC message) arrives, it goes to "proc_msg" state, processes the received message, and updates the routing table, if necessary.

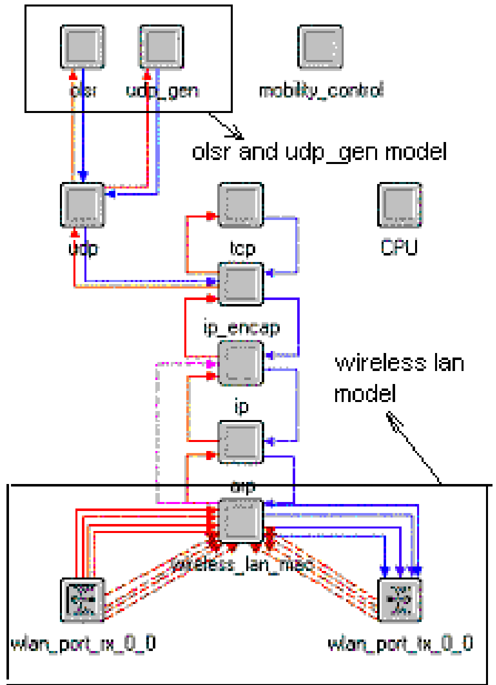


Figure 5: *OLSR Node*

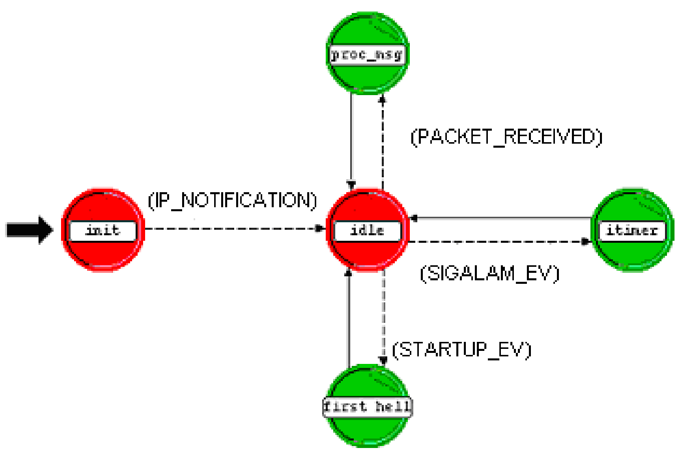


Figure 6*: OLSR Process Model*

- "udp_gen" process model

Figure 7 is the "udp_gen" process model. It generates "udp" packets, which serve as the application data packets in simulations. At the same time, it records how many "udp" packets

are received at the current node, providing a mechanism to evaluate the packet delivery ratio of a routing protocol.

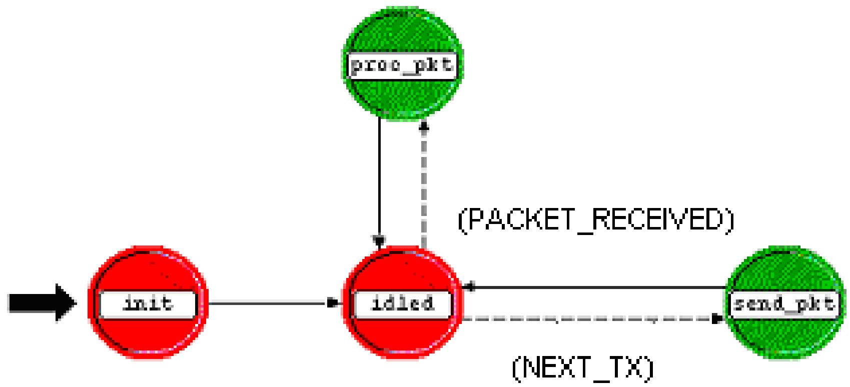


***Figure 7**: UDP_GEN Process Model*

5.2.2 QoS OLSR OPNET Model

Based on Section 3.2, the following revisions are made to develop the QoS OLSR node model:

1) Idle time calculation[1]

As mentioned before, QoS OLSR uses the media idle time to reflect the available bandwidth over a link. This task is done by modifying the standard OPNET Wireless LAN model.

Each OLSR node connects to the wireless media (see the box in lower part of Figure 5). The OPNET Wireless LAN simulation model is composed of a wireless_lan_mac process model (wireless_lan_mac), a transmitter (wlan_port_tx_0_0), a receiver (wlan_port_rx_0_0), and channel streams (the dotted line between the wireless_lan_mac and the transmitter or receiver).

If the node is sending packets, its transmitter becomes busy. If there are other nodes beginning transmission within the interference range of the current node, its receiver senses the busy media and sends a media busy signal. As the OPNET Wireless LAN model already defines functionalities to capture changes of the media, the media idle time is computed as following:

In a 0.5 second time period[2], it is recorded that how long the transmitter or receiver is busy (time between the transmitter or receiver becomes busy and then returns to idle again). Then the percentage of idle time is calculated, which is (0.5-busy time)/0.5. This is a sample of the idle time in this interval. The idle time of 10 such 0.5-second-

---

[1] In the real word, the wireless card keeps on monitoring the wireless physical medium before it sends packets, same as the implementation of the transmitter and the receiver in OPNET Wireless LAN model. So, the information that is used to calculate idle time in OPNET could also be obtained somehow through the interface of the wireless card.

[2] As in OLSR, Hello message is sent every 0.5 second, 0.5 second is used as the sampling period to reflect the traffic condition in the wireless media.

periods in a row is calculated, 10 samples of idle time over 5 seconds are obtained and arranged into a sliding window, and then its average value is calculated. When an 11th idle time sample is obtained, the 1st idle time in the sliding window is deleted, and the 11th idle time is inserted into the sliding window as the last value. See the following Figure 8 as an example:
The Wireless LAN process model continuously calculates idle time, and reports the average value to the OLSR process model.

position	0	1	2	3	4	5	6	7	8	9
Idle time	50%	50%	50%	50%	50%	50%	50%	50%	50%	50%

Original average idle time is 50%.
After new value is obtained, the updated sliding window:

the 11th value (30%) is obtained

position	0	1	2	3	4	5	6	7	8	9
Idle time	50%	50%	50%	50%	50%	50%	50%	50%	50%	30%

New average value: (50% x 9 + 30%)/10=48%

***Figure 8**: Example of How Idle Time Is Calculated*

2) Idle time propagation

As discussed in Section 3.2.2 and Section 3.2.3, the QoS OLSR versions needs to know the available bandwidth on the neighbor link to select MPRs, and the available bandwidth of the far away link to compute the routing table. As idle time should be used to calculate the available bandwidth on the links, the formats of OLSR Hello and TC messages are revised to include the idle time in it. [3]

a. Hello message: in addition to the original information such as neighbor address and neighbor link type, a node also includes its own idle time in the Hello messages. Upon receiving a Hello message from its neighbor, a node reads the neighbor idle time, and selects MPRs using the QoS MPR selection algorithm.

b. TC message: the TC message originator not only puts its own idle time in TC messages, but also piggybacks its MPR selectors' idle times, which are obtained from the Hello messages. When a node receives TC messages, it knows the idle time information of both the TC message originator and the MPR selectors, thus gets information about the links and the link bandwidth between the TC message originator and its MPR selectors. In this way, it learns the partial network topology and the bandwidth condition of that partial network, and is ready to calculate the routing table.

[3] For compatibility, it is better to introduce a new message type to propagate idle time together with the original OLSR message. However, for simplicity, for the time being, the original OLSR message is revised.

Also, QoS OLSR needs to decide when to originate a TC message. In the original OLSR, if a node detects changes in its MPR selector, it generates a new TC message to propagate the changes in the network topology. In QoS OLSR, however, changes in link bandwidth condition must also be propagated for the correct computation of the best bandwidth routes. However, because of the dynamic nature of the Ad-Hoc network, link bandwidth may change all the time. If an MPR generates a TC message as soon as it detects a bandwidth change over the link between its MPR selector and itself, there will be too many messages flooding into the network, causing extremely high overhead. So in our QoS OLSR, some "threshold" of bandwidth change is defined. If an MPR finds there is "significant bandwidth change", that is, the available bandwidth raises or drops a certain percentage, between the links of its MPR selectors and itself, it will generate a new TC message informing the whole network about the change, enabling other nodes to update their routing table reflecting such changes. There is a tradeoff in how to define the "threshold". On one hand, if the "threshold" is low, TC messages will be generated as soon as there is a small percentage change of the bandwidth. That will cause frequent generation of TC messages, introducing high overhead, although more accurate bandwidth information is obtained. On the other hand, if the "threshold" is high, TC messages will not be generated until there is a very large percentage change of the bandwidth. Thus, the overhead is reduced, but the nodes only obtain relatively inaccurate bandwidth information.

In the implementation, a node keeps on informing its original idle time in its Hello messages until the latest idle time value it obtains from the Wireless LAN process model changes above the "threshold" compared with the original idle time. In such case, the node will propagate the new idle time in the Hello message, reflecting the change in the traffic condition on the wireless media. Upon receiving such Hello message, the neighbor node re-selects MPRs according to the latest idle time information. Consequently, TC messages are generated to reflect the bandwidth change.

In the simulation, different "threshold" values are defined to compare the network performance, and analyze the "price" paid and the "profit" gained.

3) MPR selection

Based on the simulation result of the static network case in Section 4, it is found that OLSR_R2 (Section 3.2.2.2) guarantees to find the best bandwidth path while it has a lower overhead compared with OLSR_R3 (Section 3.2.2.3), which also finds the optimal bandwidth path. So in the implementation of QoS OLSR model, OLSR_R2 is used as the MPR selection mechanism.

4) Routing table calculation

As discussed in Section 3.2.3, the "Extended BF" algorithm is used to compute the routing table, as it not only finds the best bandwidth path, but the shortest path as well.

5) Idle time recording

In order to observe the routing protocols in bandwidth QoS aspect, the network bandwidth condition as well as the network topology should be recorded. As OPNET does not provide such information, a data-recording process model is developed, which takes network snapshot as the simulation goes on. Every 5 seconds[4], the data-recoding model records the positions of all nodes in the network, their idle times computed by the modified Wireless LAN model, which is discussed in 1), and the

[4] It is desirable to use even shorter time interval to obtain more accurate network information. However, because of disk space limitations, a 5 seconds interval is used here. Compared with OPNET's 9 seconds interval for exporting simulation result, 5 seconds seems to be a reasonable choice.

actual routing table each node computed. Using such information, the optimal bandwidth paths in the network snapshot can be computed, and the bandwidth difference between the routes the routing algorithms calculated and the optimal routes can be obtained.

5.3 Simulation Set Up

The following environments are defined for OPNET simulations:
Movement Space: 1000m x 1000m flat space
Number of Nodes: 50 nodes
Simulation Time: 900 seconds. Many papers that study the performance of routing protocols in Ad-Hoc network such as [14] use 900 seconds as simulation length. Besides, after 30 seconds of simulation time, the routing algorithms' performance such as packet delivery ratio and delay is rather stable. So here, 900 seconds simulation time is used for all scenarios.
Movement Model: each node randomly selects a destination in the 1000m x 1000m area, moves to that destination at a speed distributed uniformly between 0 and "maximum speed". After it reaches the destination, the node selects another destination and another speed between 0 and "maximum speed", and moves again. The model is based on the "random waypoint" model [8], but differs from the "random waypoint" model in that in "random waypoint" model, the node pauses for "pause time" seconds before it moves again, while in current movement model, nodes move continuously. In the simulation, there are 5 "maximum speed" values: 20m/s, 10m/s, 5m/s, 1m/s, and 0m/s.
Communication Model: packet sources are the udp_gen process models defined in the OLSR node model. In each simulation, there are 20 communication pairs. Each source sends 64-byte packets at a rate of 4 packets/second. So in total, 80 packets are sent each second.
OPNET Model Parameter: see Table 7.

Table 7: *OPNET Model Parameter*

OLSR Parameters	***Hello Interval***	0.5s
	TC Interval	2s
Wireless LAN Parameters	***Data Rate***	2 Mbps
	Buffer Size	256000 bits
	Retry Limit	7
	Wireless LAN Propagation Range	250 M

Routing Protocol: 4 routing protocols – Original OLSR, QoS OLSR with 20% bandwidth updating threshold (20% OLSR), QoS OLSR with 40% bandwidth updating threshold (40% OLSR), and QoS OLSR with 80% bandwidth updating threshold (80% OLSR). All the QoS OLSR algorithms use the OLSR_R2 mechanism to select MPRs, and the "Extended BF" algorithm to calculate the routing table.
For each of the 5 movement patterns (maximum speed 20m/s, 10m/s, 5m/s, 1m/s, 0m/s), 3 simulations are done for each routing protocol to test its performance. The 3 simulations differs from one another in 1) nodes starting positions, 2) communication pairs, 3) the random destinations and the uniformly distributed speed a node chooses in its movement.

VI. Performance Evaluation in OPNET

In this section, simulation results on dense network (50-nodes-network) are presented and analyzed. The network characteristic studied here is described in Section 5.3.
The results are grouped into two sets: Basic Performance and QoS Performance.

1) Basic Performance – the basic performance is the set of metrics used by most routing protocols for result comparison: “Packet Delivery Ratio” and “End to End Delay”.
   - Packet Delivery Ratio: percentage of packets that successfully reach the receiver nodes each second. Packet Delivery Ratio = average packet received per second / 80 (the total packet sent per second) * 100%
   - End to End Delay: the average time between a packet being sent and being received
2) QoS performance – the metrics that relate to the bandwidth QoS routing studied in this paper: “Error Rate” and “Bandwidth Difference”.
   - Error Rate: the percentage of times the routing algorithms do not find the optimal bandwidth path.
   - Bandwidth Difference: the average difference between the optimal bandwidth and current bandwidth in percentage, which is less than the optimal one, found in routing algorithms. Result = average of (bandwidth on optimal path - bandwidth on route computed)/bandwidth on optimal path, when the optimum routes are not found.

For all simulation results presented in this chapter, two kinds of data are shown: one is the average result, which is listed in the upper part of the table cell; the other is the width of the confidence interval, calculated with 95% confidence, which is in the lower part of the table cell.

6.1 Basic Performance

Table 8 shows the Basic Performance results of the 4 OLSR routing algorithms (QoS 20%, QoS 40%, QoS 80%, original) for 5 movement patterns (maximum speed: 20m/s, 10m/s, 5m/s, 1m/s, 0m/s).
(Here, PK Delivery Ratio=Packet Delivery Ratio; E-to-E Delay=End to End Delay)

Table 8: *Packet Delivery Ratio and End-to-End Delay Comparison for 50-Node-Network Scenario*

	Speed: 20m/s		**Speed: 10m/s**		**Speed: 5m/s**	
	PK Delivery Ratio	***E-to-E Delay (ms)***	***PK Delivery Ratio***	***E-to-E Delay (ms)***	***PK Delivery Ratio***	***E-to-E Delay (ms)***
QoS 20%	66.89% 2.96%	24.92 2.64	75.71% 0.63%	14.82 2.31	84.66% 1.74%	9.55 1.11
QoS 40%	67.59% 1.39%	20.16 2.83	79.21% 4.63%	13.70 7.19	88.05% 2.68%	10.43 1.89
QoS 80%	72.05% 5.20%	24.70 23.54	79.91% 4.30%	18.88 17.33	89.46% 3.95%	7.78 4.93
Original	75.75% 2.91%	8.58 3.16	82.30% 3.28%	5.73 0.64	87.81% 1.20%	5.28 1.54

	Speed: 1m/s		Speed: 0m/s	
	PK Delivery Ratio	***E-to-E Delay (ms)***	***PK Delivery Ratio***	***E-to-E Delay (ms)***
QoS 20%	90.89% 2.28%	9.20 4.98	98.15% 3.16%	13.05 8.16
QoS 40%	94.31% 2.14%	9.84 5.16	99.53% 0.48%	9.04 7.09
QoS 80%	93.44% 7.28%	7.09 6.72	97.58% 6.90%	8.11 5.77
Original	96.34% 0.49%	4.67 1.13	98.54% 1.00%	5.88 2.52

6.1.1 Packet Delivery Rate

Figure 9 shows the comparison of the packet delivery ratio the 4 algorithms achieve under different movement patterns.

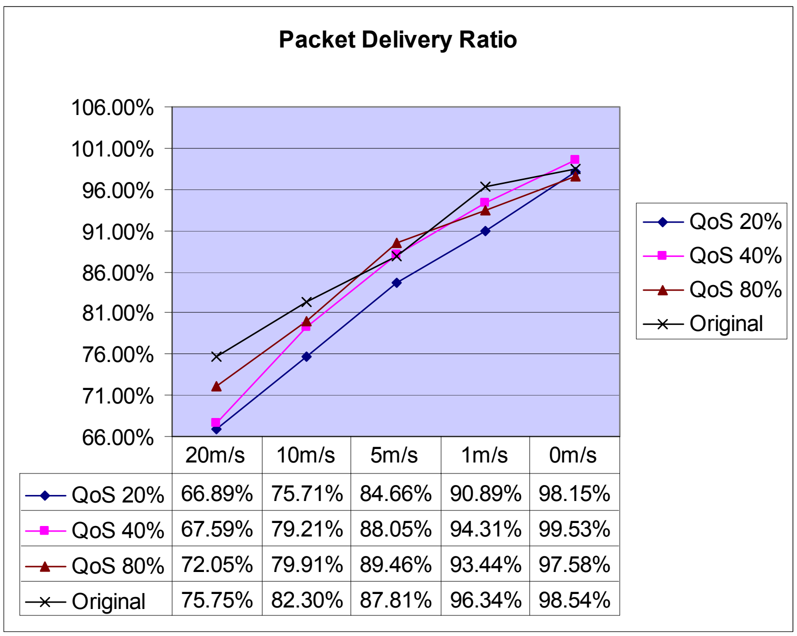


***Figure 9**: Comparison of Packet Delivery Ratio for 4 OLSR Algorithms in 50-Nodes-Network*

From high movement (maximum speed 20m/s) to low movement (maximum speed 0m/s), packet delivery ratio for all algorithms rises continuously. It is easy to understand. With the lower movement, the established links between the nodes have a lower probability to break, thus, there are less stale routes in the node routing tables, which results in a higher ratio for correct packet delivery. However, in the 4 OLSR algorithms, the original OLSR outperforms the other 3 QoS version of OLSR algorithms in packet delivery, especially at high mobility (maximum speed: 20m/s). There are two reasons behind it:

a. High Overhead: As mentioned in Section III, the original OLSR protocol concentrates on how to reduce the overhead, and tries to minimize the MPR sets to reduce the TC messages flooding into the network. However, the QoS versions of OLSR attempt to select the best bandwidth path, so in their MPR selection mechanism, they select neighbors with high idle time as MPR, resulting in a larger MPR set than the original OLSR protocol. So more TC messages are generated and relayed into the network by QoS OLSR versions. The following table (Table 9) and Figure 10 and 11 show the average TC messages generated or relayed by all MPRs in the network (in packets and in kbps) for the 4 algorithms:

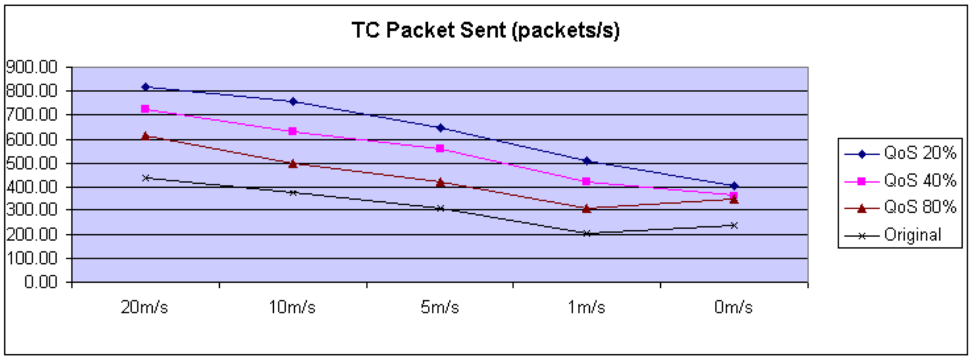


Figure 10: *TC Packet Sent in Packet/S*

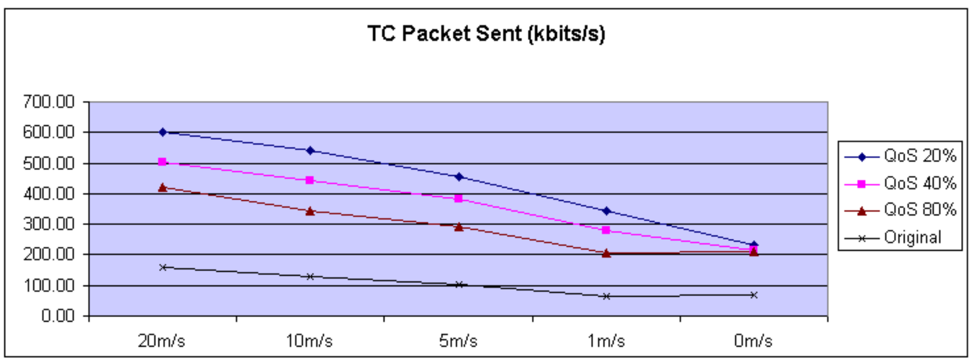


Figure 11: *TC Packet Sent in Kbps*

Table 9: *Comparison of TC Message Sent for 4 OLSR Algorithms in 50-Node-Network Scenario*

Algorithm	Speed: 20m/s		Speed:10m/s		Speed: 5m/s		Speed: 1m/s	
	Packets/s	*Kbps*	*Packets/s*	*Kbps*	*Packets/s*	*Kbps*	*Packets/s*	*Kbps*
QoS 20%	816.00 16.38	600.99 31.64	755.62 45.86	543.67 24.49	649.36 51.02	458.36 55.41	510.88 110.21	341.16 71.65
QoS 40%	726.74 29.36	501.94 18.04	630.68 18.52	442.19 23.03	558.60 15.30	379.11 21.75	423.28 122.43	279.17 67.55
QoS 80%	614.09 48.42	423.48 32.44	497.39 54.88	339.89 8.27	424.62 59.14	289.51 29.35	306.99 90.25	205.76 48.53
Original	439.15 19.19	156.77 11.53	372.13 57.29	128.55 19.23	305.48 14.11	102.45 8.04	200.51 75.23	64.78 21.68

Algorithm	Speed: 0m/s	
	Packets/s	*Kbps*
QoS 20%	406.71 57.64	232.14 55.68
QoS 40%	362.46 17.74	214.85 24.74
QoS 80%	347.48 30.66	209.53 33.04
Original	236.66 102.56	69.62 28.81

From the table and the figures, it can be seen that for all algorithms, there are fewer TC messages sent at lower movement than at higher movement. This is because at lower movement, less TC messages are generated to reflect topology changes. Also, 20% OLSR has the highest number of TC messages generated and relayed, while the original OLSR protocol has the least number of TC messages. Under the same speed, the difference of TC messages sent between the original OLSR protocol and the 3 QoS OLSR versions comes from three aspects:

1) The original OLSR protocol only generates TC messages to reflect topology change, while QoS OLSR versions also need to generate TC messages to reflect bandwidth change; with a lower bandwidth update threshold, more TC messages are generated to reflect bandwidth change, causing the highest overhead in 20% OLSR

2) The average TC packet length in QoS OLSR versions is larger then that of the original OLSR protocol, as in the QoS OLSR versions, TC messages not only include the addresses of the MPR selectors, but also their idle times.

3) QoS OLSR versions have larger MPR sets than the original OLSR protocol, so more TC messages are generated and relayed by the larger MPR sets. Among the QoS OLSR algorithms, 20% OLSR may select more MPRs than 40% and 80% OLSR. The following is the explanation:

As mentioned in Section 5.2.2, in QoS OLSR, a node continues announcing its original value of idle time in the Hello messages until its own idle time rises or drops over a certain threshold; then, the node announces its new idle time. Also, nodes select MPRs based on the link bandwidth, in other word, neighbors' idle time. Based on the way the idle time is calculated, at the beginning of the simulation, the whole wireless media is idle, so all nodes' initial idle times are 100%.

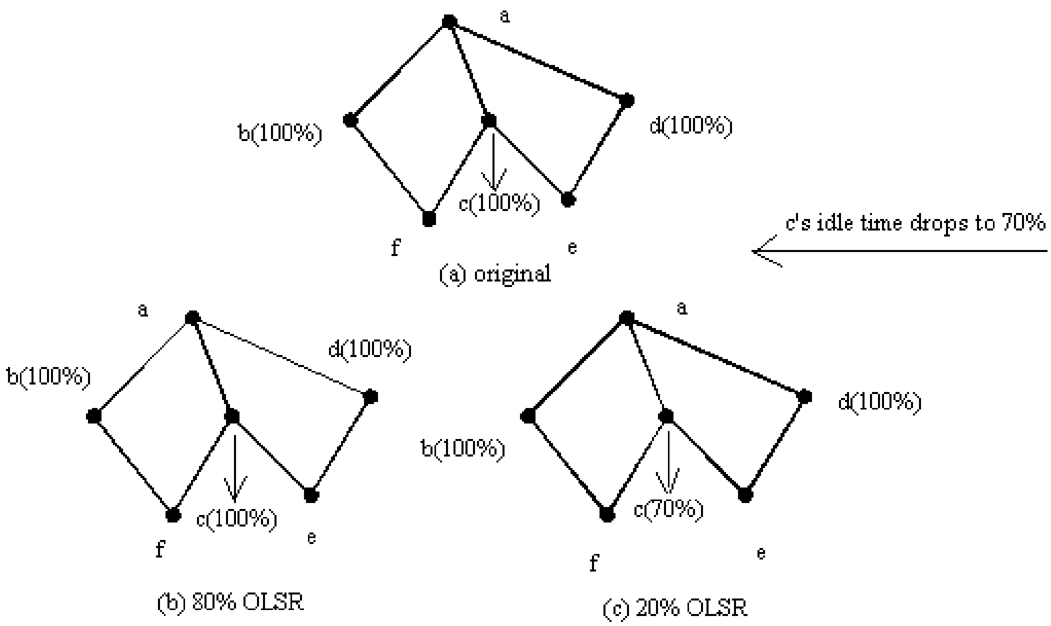
With a low idle time updating threshold such as 20%, the neighbor idle times are more diverse than with high idle time updating thresholds such as 40% or 80%. Recall that if the neighbor idle times are the same, a node selects the one that covers most un-reached 2-hop-neighbors as MPR. Otherwise, it keeps on selecting neighbors with higher idle time as MPRs until all the 2-hop-neighbors are covered. So a neighbor set with more diversity of idle times may result in a higher number of MPRs, see Figure 12 as an example.

From Figure 12, it can be seen that if a node's neighbor set has a high diversity of idle time values, the node may have a higher probability to select more MPRs, depending on the network topology.

With the possibly larger MPR set, more TC messages are generated and relayed by 20% OLSR than 40% OLSR and 80% OLSR.

The overhead (TC messages sent) in the fixed network differs a little from the above observation. The overhead for 20% and 40% OLSR still keeps the same trend as before – the number of TC messages sent in the fixed network is less than for a maximum speed of 1m/s. However, more TC messages are sent in 80% OLSR and the original OLSR for movement 0m/s than 1m/s. The explanation is that in the fixed

network, where there is no node movement, in the original OLSR, TC messages are sent regularly at 2s interval. So the TC message overhead is solely related to the number of MPRs in the network, which depends on the network topology. The network topology does not change during the simulation, and only 3 simulations are run for each algorithm under each movement pattern. For the fixed network case, actually, just 3 samples of network "snapshots" are taken, which may not be enough to give an exact result. The 80% OLSR may have the same problem in the fixed network, as with a large threshold for bandwidth updates, TC messages sent in the network may mainly be decided by the number of MPRs in the network, which does not change often in the static network. Considering the confidence interval, there is a large overlap for the value shown for 1m/s and 0m/s scenario, which means there is not too much overhead difference between the extremely low movement scenario (1m/s) and the no movement scenario (0m/s), which is consistent with our basic explanation.



In (a), initially, all the neighbors of a -- b, c, and d have the same idle time: 100%. So based on th QoS OLSR, node a select c, which covers both of a's 2-hop-neighbors d and e, as MPR.
Then, c's idle time drops to 70%.
In (b) 80% OLSR, as the changes of c's idle time is 30%, which is less than its threshold of 80%, c still propogates its idle time as 100%. So from a's point of view, b, c, and d have the same idle time. It still selects c as MPR to cover e and f. So in 80% OLSR, a only has one MPR -- c.
In (c) 20% OLSR, as the change in idle time is more than 20% threshold, c updates its idle time information in its Hello message. Learning that c's idle itme is 70%, less than the other neighbors, a reselects its MPRs. Now, both b and d are a's MPRs to cover e and f. So in 20% OLSR, a has two MPRs -- b and d.

Figure 12*: MPR Selection in QoS OLSR with Different Thresholds*

With higher overhead introduced into the network, especially for the 20% OLSR at higher movement, the wireless media is extremely busy, imposing a negative impact on the packet delivery rate for QoS versions of OLSR.

b. Incorrect Routing Table: besides the delay of topology updating information, which causes stale routes in the routing table, the following Figure 13 shows another typical scenario that causes incorrect topology information:

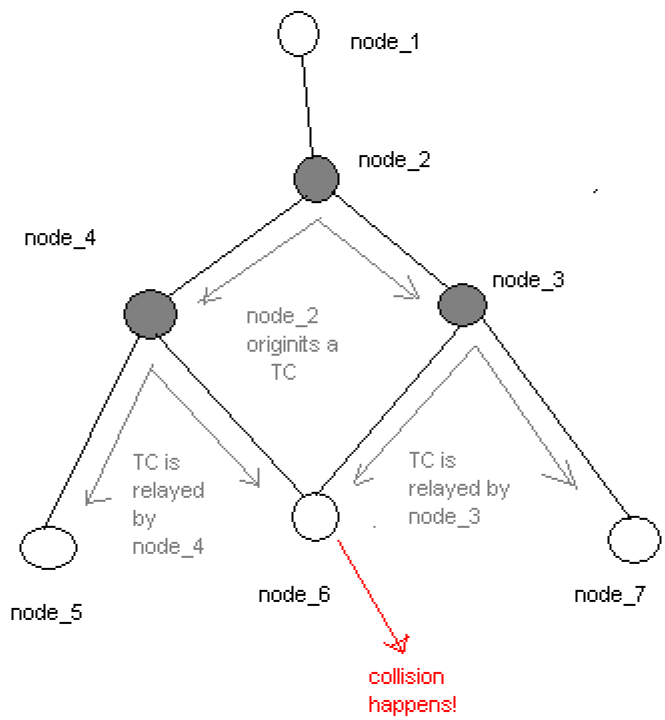


Figure 13: *An Example for TC Packet Collisions at the Physical Layer*

In Figure 13, based on the original OLSR algorithm, node_2 is selected as MPR by node_1, and generates a TC message advertising that there is a link between node_1 and node_2. Node_3 and node_4 are all MPRs of node_2, so they both relay the TC message. Suppose at that time, the wireless media is idle, node_3 and node_4 relay that TC message immediately, most probably at the same time. As a result, the TC messages collide at node_6, and node_6 does not know that it can reach node_1 through node_2. From this example, it can be seen that if there are overlapped two hop neighbors covered by multiple MPRs, there is a high probability that TC packets collide at these neighbors, causing problems in routing table calculation. This problem happens in all 4 OLSR algorithms. But because of the different MPR selection mechanism, the QoS OLSR algorithms have more overlapped two hop neighbors than the original OLSR protocol, causing more TC message collisions.

How does the above two reasons impact on the packet delivery ratio of the Ad-Hoc routing protocol? Table 10 shows the breakdown of unsuccessfully delivered packets.

In Table 10, besides the information about "TC sent", the following metrics are also presented:

- Packet Un-delivered: the average number of udp data packets that do not reach the destination in each second. Packet Un-delivered=(1-Packet Delivery Ratio)*80, as total packets sent by the network in each second is 80
- IP PK Dropped: average number of packets dropped at the IP layer each second. This is an OPNET build-in metric, which represents the number of packets dropped at the IP layer because there is no entry about the destination in the IP routing table. (The IP routing table does not know the next hop for a certain destination.)
- Control Bad PK: the average number of TC or Hello packets that experience collision at the wireless_lan_mac layer each second. This is an important metric to reflect the correctness of the routing table built by the routing algorithm. As TC messages include information about the network topology, the collision of TC messages means that the node could not get the updated topology information abut the remote part of the network, and could not correctly build the routing table, which will result in packet dropping in either the IP layer (the remote node is reachable, but the routing table does not include such entry) or the Wireless LAN layer (a packet is sent to a node out of the transmission range based on a stale route in the routing table. As the sending node cannot receive the Ack, it keeps on retransmission until the retry limit is passed and the packet is dropped.)
- Data Bad PK: the average udp data packets that experience collision at the wireless media in one second. A data packet experiencing collision doesn't necessarily mean it can not be correctly delivered, as a data packet can be re-transmitted for 7 times before it is dropped.
- WLAN PK Dropped: average number of packets dropped at the wireless_lan_mac layer. This is also an OPNET build-in metric. There are two reasons for packets dropped in this layer: 1) the overflow of higher layer buffer, and 2) failure of all retransmissions until retry limit (7). Both reasons are related to the control overhead/TC messages sent —on one hand, if there are many control packets, the wireless media is very busy, the probability that the data packet experiences collision is high, and the probability that it is dropped because of all retry chances are used up is also high; on the other hand, too many packets waiting to be processed also causes the overflow of the higher layer queue.

Let us take the 20m/s scenario as an example. Referring to Figure 16, it can be seen that because the original OLSR protocol has the smallest MPR set, it has the smallest number of control packet collisions (see the category of "Control Bad PK), resulting in the smallest number of packets dropped at the IP layer ("IP PK Dropped"). Also, it has the smallest number of packets dropped at the Wireless LAN ("WLAN PK Dropped"). Compared with the QoS OLSR versions, its low overhead results in a relatively less busy wireless media, reducing the possibility of overflow of higher layer queue and packet collisions.

Among the 3 QoS OLSR algorithms, as discussed before, 20% OLSR may have the largest MPR set, because with the more accurate link bandwidth information, it may select more MPRs than the other two QoS algorithms, resulting in more overlapped two hop neighbors. This is why the 20% OLSR has the largest number of TC message collisions, and the largest number of packets dropped at the IP layer. With the same reason, the 80% OLSR gets the most correct information about the network topology and has the lowest number of packets dropped at the IP layer.

Table 10: *Where Are the Unsuccessfully Delivered Packets Dropped?*

Speed	Algorithm	TC Sent (pks/s)	Packet Undelivered (pks/s)	IP PK Dropped (pks/s)	Control Bad PK (pks/s)	Data Bad PK (pks/s)	WLAN PK Dropped (pks/s)
20m/s	QoS 20%	816.00	26.49	6.15	2481.03	29.65	20.23
	OLSR	16.38	2.37	2.40	235.82	2.69	1.50
	QoS 40%	726.74	25.93	4.12	2064.78	25.32	21.82
	OLSR	29.36	2.83	0.93	86.87	3.99	0.64
	QoS 80%	614.09	22.36	1.84	1767.88	17.22	20.39
	OLSR	48.42	4.16	0.38	100.65	4.29	4.17
	Original	439.15	19.40	0.64	1285.95	11.39	18.67
	OLSR	19.19	2.33	0.33	120.24	0.87	2.38
10m/s	QoS 20%	755.62	19.43	5.24	2252.42	28.54	14.12
	OLSR	45.86	2.31	0.72	82.66	0.68	1.21
	QoS 40%	630.68	16.63	3.32	1891.44	20.79	13.26
	OLSR	18.52	3.70	2.35	135.10	7.10	1.38
	QoS 80%	497.39	16.07	1.82	1454.60	17.35	14.18
	OLSR	54.88	3.44	0.79	112.75	3.37	3.75
	Original	372.13	14.16	2.09	1087.10	9.85	12.00
	OLSR	57.29	2.62	1.09	117.35	3.49	1.51
5m/s	QoS 20%	649.36	12.27	4.30	1920.89	27.34	7.93
	OLSR	51.02	1.39	0.65	249.25	2.87	0.73
	QoS 40%	558.60	9.56	2.03	1605.54	19.30	7.50
	OLSR	15.30	2.14	1.35	116.99	7.75	1.07
	QoS 80%	424.62	8.43	1.60	1248.21	12.94	7.47
	OLSR	59.14	3.16	1.57	51.94	3.04	3.80
	Original	305.48	9.75	0.62	818.62	11.67	9.11
	OLSR	14.11	0.96	0.72	153.38	1.07	0.53
1m/s	QoS 20%	510.88	7.29	4.96	1435.72	33.57	2.32
	OLSR	110.21	1.82	0.49	188.93	20.67	1.28
	QoS 40%	423.28	4.55	2.13	1172.97	20.87	2.40
	OLSR	122.43	1.71	0.56	198.05	19.26	1.50
	QoS 80%	306.99	5.25	3.01	1129.49	13.43	2.22
	OLSR	90.25	5.82	5.28	189.08	15.11	1.54
	Original	200.51	2.93	0.75	476.91	9.87	2.17
	OLSR	75.23	0.39	1.39	100.98	5.41	1.01
0m/s	QoS 20%	406.71	1.48	1.37	829.73	33.35	0.10
	OLSR	57.64	2.53	2.53	178.96	17.15	0.04
	QoS 40%	362.46	0.38	0.36	731.52	21.34	0.01
	OLSR	17.74	0.38	0.36	89.79	23.99	0
	QoS 80%	347.48	1.94	1.93	718.25	19.60	0
	OLSR	30.66	5.52	5.49	113.65	28.92	0
	Original	236.66	1.17	1.16	355.19	13.07	0
	OLSR	102.56	0.80	5.06	212.67	7.06	0

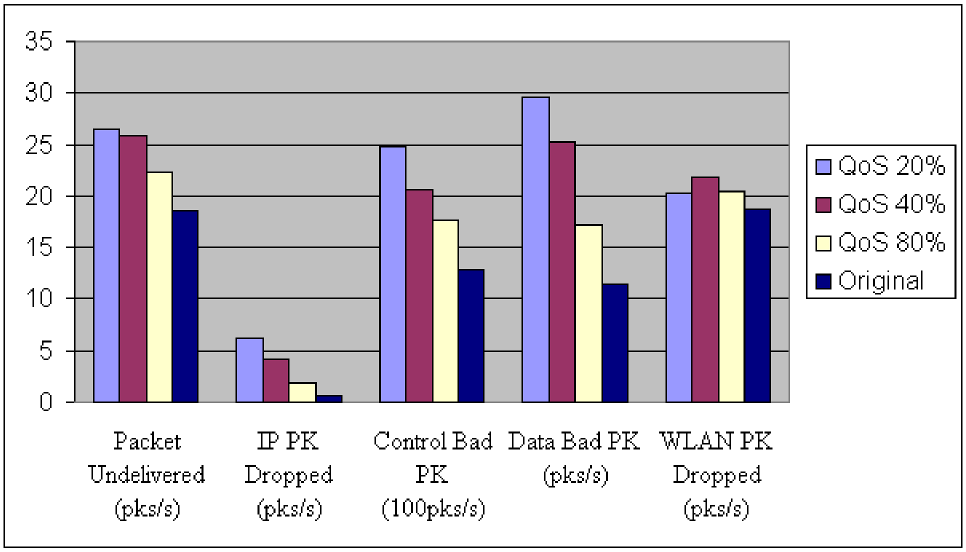


Figure 14: *Relationship between Packets Undelivered and Packets Dropped at Different Layers (20m/s)*

The packets dropped in the Wireless LAN of the 3 QoS OLSR algorithms, however, are very close, although the 20% OLSR introduces much more control traffic into the network. To explain this phenomenon, recall that the route computation of QoS OLSR always directs data traffic to the routes with higher bottleneck bandwidth, which means, ideally, the data traffic in the 20% OLSR always chooses a route that is less busy, causing relatively low overflow compared with its high overhead level.

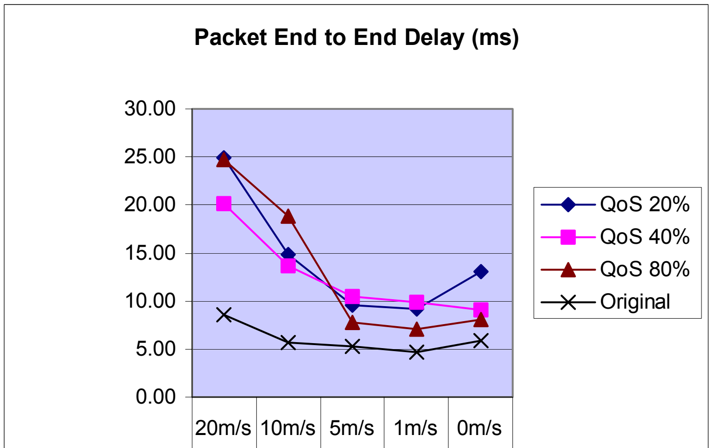
The behavior of the 4 OLSR algorithms in other movement patterns can be analyzed similarly. Note that at lower speed scenarios (5m/s, 1m/s, and 0m/s), the packet delivery ratio for the QoS OLSR versions is close to the original OLSR protocol. At low movement, the control overhead is reduced for all algorithms, resulting in a relatively less busy wireless media. Consequently, the additional overhead introduced by QoS OLSR versions will not have as negative an effect on the packet delivery as in high movement scenarios.

From the data collected, it can also be concluded that the main reason for the packet delivery ratio difference among the 4 OLSR algorithms is the correctness of routing tables calculated, as the difference in the “IP PK Dropped” among all 4 algorithms is almost the same as the difference in “Packet Un-delivered”.

6.1.2 End-to-End Delay

Based on Table 8, Figure 15 shows the End-to-End Delay for each algorithm under each movement pattern.

Basically, for all movement patterns, the original OLSR has the lowest delay. It is easy to understand. As the original OLSR has the lowest overhead, its network is the least congested, resulting in the least delay. Also, the original OLSR algorithm always computes the shortest hop path, while the QoS OLSR versions may compute longer paths because they target on the best bottleneck bandwidth path, which also affects the end-to-end delay of the data packets.



	20m/s	10m/s	5m/s	1m/s	0m/s
QoS 20%	24.92	14.82	9.55	9.20	13.05
QoS 40%	20.16	13.70	10.43	9.84	9.04
QoS 80%	24.70	18.88	7.78	7.09	8.11
Original	8.58	5.73	5.28	4.67	5.88

***Figure 15**: Comparison of End-To-End Delay of Data Packets for 4 OLSR Algorithms in 50-Nodes-Network*

For the three QoS OLSR algorithms, at higher movement speed (20m/s and 10m/s), the 80% threshold QoS OLSR has a higher delay, while at lower movement speed (5m/s, 1m/s and 0m/s), its delay is close to the original OLSR. To analyze this phenomenon, recall that the 80% threshold QoS OLSR has the most inaccurate bandwidth information of the network, which means that the routing algorithm may select a route that is still relatively congested. At higher movement, all the QoS OLSR algorithms have higher overhead because of the frequent updates due to topology change (see Table 9 and Figures 10, 11), making the network congested. Working on the already congested networks, 20% QoS OLSR and 40% QoS OLSR do a better job in directing the traffic to the less congested routes, resulting in the lower packet delay. However, at lower movement speed, there are much less topology updates, so the more frequently sent bandwidth update messages in 20% and 40% OLSR tend to make the network busy, resulting in a larger delay than the 80% OLSR.

Again, for all algorithms, the delay is reduced with speed dropping from 20m/s to 1m/s, with the exception for a speed of 0m/s. The packet delay in static networks is higher than the delay in networks with 1m/s movement. In the static and low movement network, because of the low control overhead, packet delay may mainly be affected by the length of the path the packet travels. In the static network, because there is no movement, there is a higher probability that the communication pairs are far away, which does not change in the simulation time. In the 1m/s scenario, nodes change positions, resulting on average in a shorter path length than in the static network. That is why the delay in the 1m/s network is lower than that in the static network.

6.2 QoS Performance

In this sub-section, the QoS performance of the 4 OLSR routing algorithms is discussed.

Figure 16, 17, and Table 11 show the "Average Difference" and "Error Rate" among the 4 algorithms under different movement patterns.

Table 11: *QoS Performance Comparison of 4 OLSR Algorithms in 50-Nodes-Network*

Speed	Algorithm	Bandwidth Difference	Error Rate
20m/s	***QoS 20% OLSR***	10.17%	18.19%
		1.53%	0.41%
	QoS 40% OLSR	15.41%	26.71%
		0.98%	4.98%
	QoS 80% OLSR	25.80%	37.17%
		2.07%	2.77%
	Original OLSR	28.96%	43.29%
		0.60%	2.22%
10m/s	***QoS 20% OLSR***	9.89%	17.50%
		0.52%	0.62%
	QoS 40% OLSR	15.57%	26.35%
		1.18%	2.42%
	QoS 80% OLSR	25.57%	39.65%
		0.18%	3.41%
	Original OLSR	30.97%	43.55%
		2.86%	0.38%
5m/s	***QoS 20% OLSR***	9.41%	18.25%
		0.78%	0.83%
	QoS 40% OLSR	14.26%	26.69%
		1.64%	1.92%
	QoS 80% OLSR	25.63%	38.70%
		0.80%	3.60%
	Original OLSR	30.33%	46.35%
		2.45%	2.28%
1m/s	***QoS 20% OLSR***	9.19%	18.76%
		1.80%	2.33%
	QoS 40% OLSR	14.61%	28.98%
		0.82%	4.43%
	QoS 80% OLSR	21.12%	40.64%
		3.13%	3.13%
	Original OLSR	27.51%	47.68%
		1.09%	3.20%
0m/s	***QoS 20% OLSR***	8.98%	13.37%
		0.58%	9.60%
	QoS 40% OLSR	13.18%	26.24%
		3.07%	23.40%
	QoS 80% OLSR	18.99%	43.65%
		2.74%	14.34%
	Original OLSR	19.54%	53.28%
		5.17%	16.17%

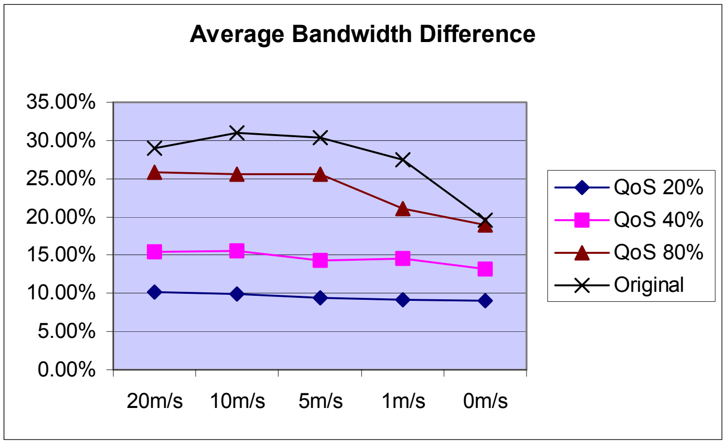


Figure 16: *Comparison of Average Bandwidth Difference for 4 OLSR Algorithms in 50-Nodes-Network*

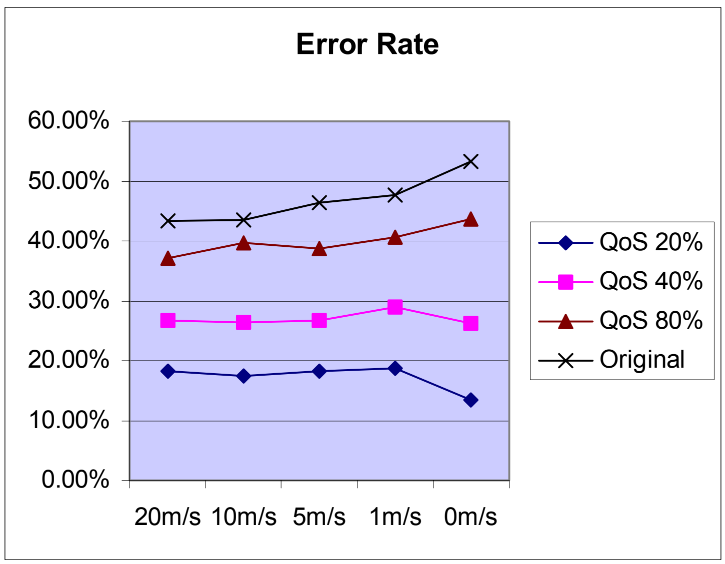


Figure 17: *Percentage of Time the 4 OLSR Algorithms Do Not Find the Optimal Bandwidth Route in 50-Nodes-Network*

All QoS OLSR outperform the original OLSR in both the “Error Rate” and “Bandwidth Difference”. Among the QoS OLSR algorithms, 20% OLSR updates the bandwidth condition most frequently, introducing the highest overhead, but gets the most accurate bandwidth information. So the routes it calculates are closest to the optimal routes.

The 40% and 80% OLSR, however, update bandwidth information less frequently, introducing less overhead, but their QoS performances are not as good as that of 20% OLSR.

In the above, the results for “Bandwidth Difference” and “Error Rate” of each algorithm are calculated based on its own network conditions – the bandwidth difference between the routes the routing algorithm calculated and the optimal paths in the network in which the routing algorithm works are presented. However, because the QoS OLSR versions introduce more overhead than the original OLSR protocol, the networks in which the QoS OLSR versions

work may have worse overall available bandwidth than that of the original OLSR algorithm. So one may question if the QoS OLSR versions really improve the route bandwidth condition. To clarify, the average available bandwidth over the routes the routing algorithms computed is presented as follows:

(Please note, as in our model, available bandwidth = maximum bandwidth x idle time in percentage, here, the available bandwidth is shown as percentage of idle time.)

To calculate the average available bandwidth on the routes the routing algorithms calculate, first, the average optimal routes bandwidth is obtained, see Table 12.

Table 12: *Available Bandwidth on the Optimal Paths in the Network the Routing Algorithm Works (Measured as Idle Time)*

Algorithm	20m/s	10m/s	5m/s	1m/s	0m/s
QoS 20% OLSR	77.68% 4.18%	80.93% 6.12%	82.29% 4.92%	84.69% 3.55%	89.73% 0.46%
QoS 40% OLSR	82.23% 7.20%	84.92% 1.60%	86.29% 2.45%	87.46% 1.53%	90.17% 0.97%
QoS 80% OLSR	78.17% 18.16%	84.27% 5.20%	87.17% 1.26%	90.08% 2.54%	92.34% 2.48%
Original OLSR	87.07% 5.37%	87.28% 3.00%	90.63% 4.03%	91.14% 1.72%	93.08% 0.43%

The above results are consistent with our former analysis: The lower the moment speed, the less the overhead all the OLSR algorithms introduce into the network. So from speed 20m/s to 0m/s, the optimal bandwidth conditions for all the OLSR algorithms rise continuously. The original OLSR algorithm has the least overhead, so the network where it works always has the best bandwidth condition. Compared with 80% OLSR, 40% OLSR evenly directs traffic throughout the network, so under high movement (speed 20m/s, and 10m/s) where the wireless media are rather busy, 40% OLSR has better optimal bandwidth routes than that of the 80% OLSR, although it has more overhead than 80% OLSR. Under low movement (speed 5m/s, 1m/s, and 0m/s), the added overhead of 40% OLSR has a negative effect on the network bandwidth condition, thus the 40% OLSR has less optimal bandwidth than 80% OLSR. As the 20% OLSR has the highest overhead, its optimal bandwidth routes have the lowest available bandwidth.

Then, the actual average available bandwidths on the routes the routing algorithms compute is calculated.

The actual average available bandwidth the routing algorithms calculated

= the available bandwidth on the optimal paths x

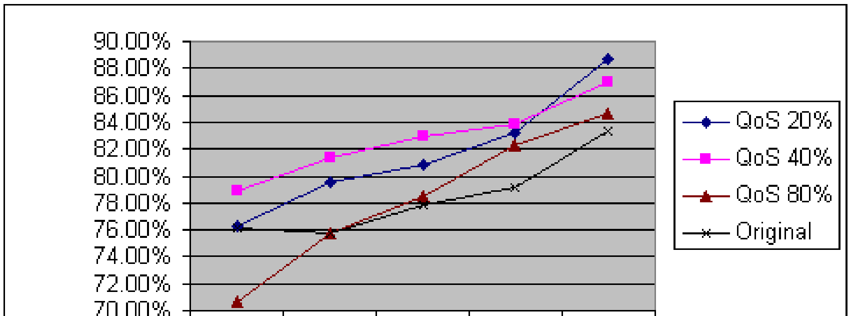
((1- "Bandwidth Difference") x "Error Rate") + (1- "Error Rate"))

= the available bandwidth on the optimal paths x

(1- "Bandwidth Difference" x "Error Rate")

Using the "Bandwidth Difference" and "Error Rate" values in Table 11, the result for actual average available bandwidth the routing algorithms calculated is shown in Figure 18[5].

[5] As the calculation includes multiply operation, the width of the confidence interval for the "actual average bandwidth the routing algorithms calculated" is not calculated.



	20m/s	10m/s	5m/s	1m/s	0m/s
QoS 20%	76.24%	79.53%	80.87%	83.23%	88.65%
QoS 40%	78.84%	81.43%	83.00%	83.87%	87.05%
QoS 80%	70.67%	75.73%	78.52%	82.35%	84.69%
Original	76.15%	75.71%	77.89%	79.18%	83.39%

***Figure 18**: Average Available Bandwidth (in Idle Time) on the Routes the 4 OLSR Algorithms Compute (50-Nodes-Network)*

From the above, it can be seen that although the QoS OLSR introduces more overhead into the network, the rout it computes still have better available bandwidth than the original OLSR. In movement patterns with maximum speed 20m/s, 10m/s, 5m/s, and 1m/s, among all the OLSR algorithms, the 40% OLSR always computes the route with the best available bandwidth, as it has less overhead than 20% OLSR and more accurate bandwidth information than 80% OLSR. In the fixed network case, because of few topology updates, all the algorithms have low overhead. Thus, 20% OLSR find the routes with highest bandwidth, for it has the most accurate bandwidth information.

From the above results, it is convincing that the QoS OLSR versions do achieve bandwidth improvement over the original OLSR algorithm.

6.3 Analyzing Simulation Results with Confidence Interval

The above Section 6.1 and Section 6.2 analyze the simulation result based on the average value. This section looks at the confidence intervals to see which sets of the performance of the 4 OLSR algorithms are statistically significant and which are not.

1) Packet Delivery Ratio

Figure 19 shows the comparison of the Packet Delivery Ratio for all the 4 OLSR algorithms under all movement patterns. In each graph, the value of the upper and lower end of the vertical line is the upper and lower bound of the Packet Delivery Ratio of each OLSR algorithm; the points which are connected by the line crossing the graph are the average values. If there is no overlap of the range of the confidence interval, it can be said that the algorithms' difference in performances is statistically significant.

From the graphs, it can be seen that in high movement patterns (20m/s and 10m/s), the observed Packet Delivery Ratio performance improvement of the original OLSR protocol over the 20% OLSR and 40% OLSR is statistically significant. However, with low movement patterns (5m/s, 1m/s, and 0m/s), the 4 algorithms' difference in performance is not statistically different. This is consist with our analysis in Section 6.1.1 – with higher movement speed, the added overhead of 20% and 40% OLSR have a negative effect on the Packet Delivery Ratio, because the networks are already congested with frequently topology update message.

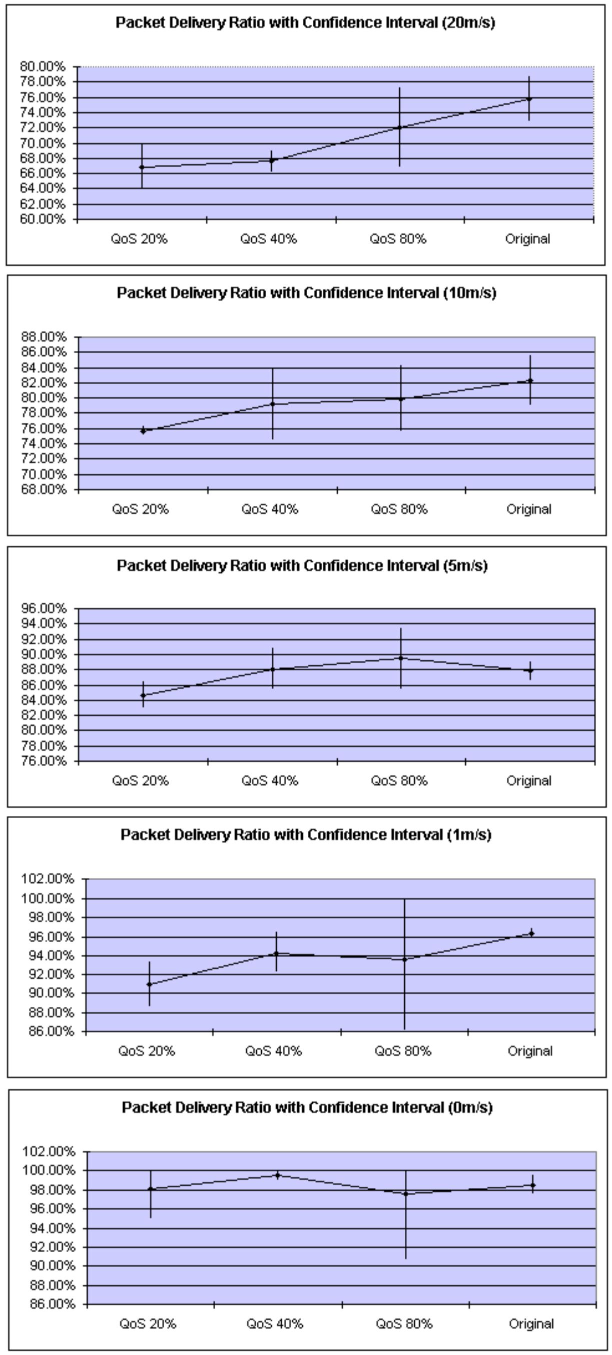


***Figure 19**: Packet Delivery Ratio Comparison with Confidence Intervals*

2) End-to-End Delay

Figure 20 shows the End-to-End Delay with confidence intervals. All the values shown have the unit “ms”.

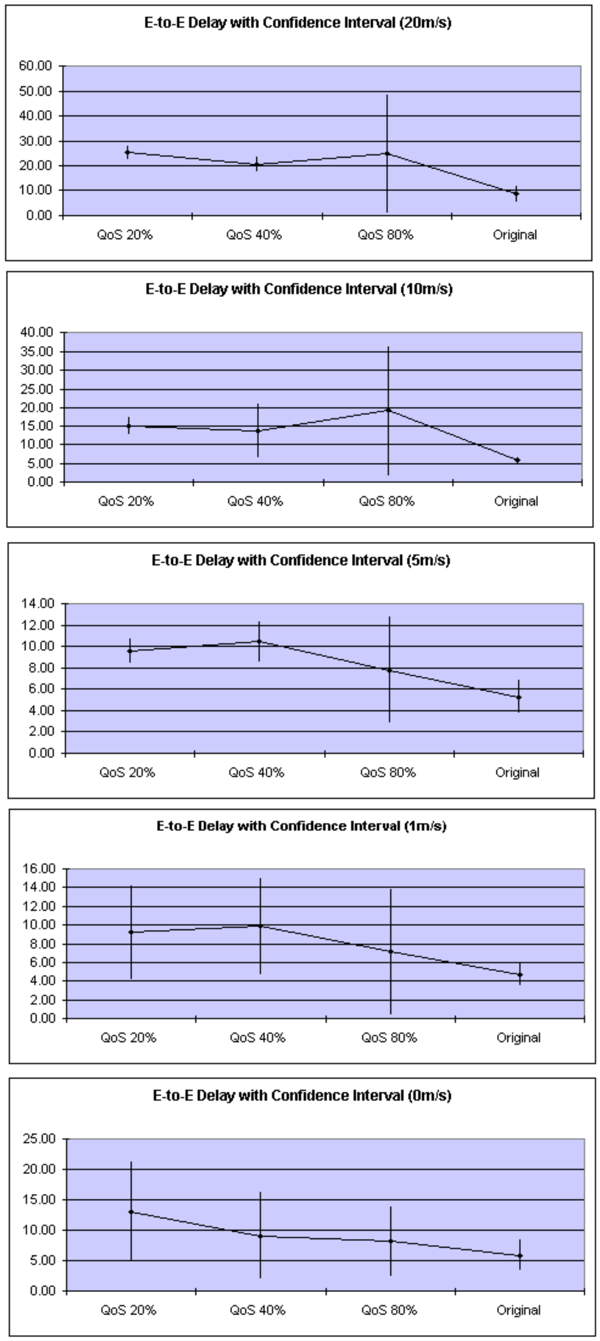


***Figure 20**: End-To-End Delay Comparison with Confidence Intervals*

In movement patterns 20m/s, 10m/s, and 5m/s, the confidence intervals for 20% OLSR and 40% OLSR have no overlap with the confidence interval for original OLSR, which means the difference of End-to-End Delay performance between 20% OLSR and 40% OLSR and the original OLSR is statistically significant. Although the range of the confidence interval of the End-to-End delay in 80% OLSR overlaps the original OLSR, its large interval means that the End-to-End Delay performance of 80% OLSR is highly variable. On the whole, because of

the large overhead that the QoS OLSR algorithms introduce into the network, they result in a higher delay than the original OLSR.

In the static networks (speed 0m/s), the End-to-End Delay performance of the 4 algorithms is rather close. Because of the low overhead in the static network, the delay may mainly be decided by the hop counts of the routes the 4 algorithms computed, which may not significantly differ from one another.

3) QoS Performance

Figure 21 presents the QoS Performance comparison for all 4 OLSR algorithms with confidence interval.

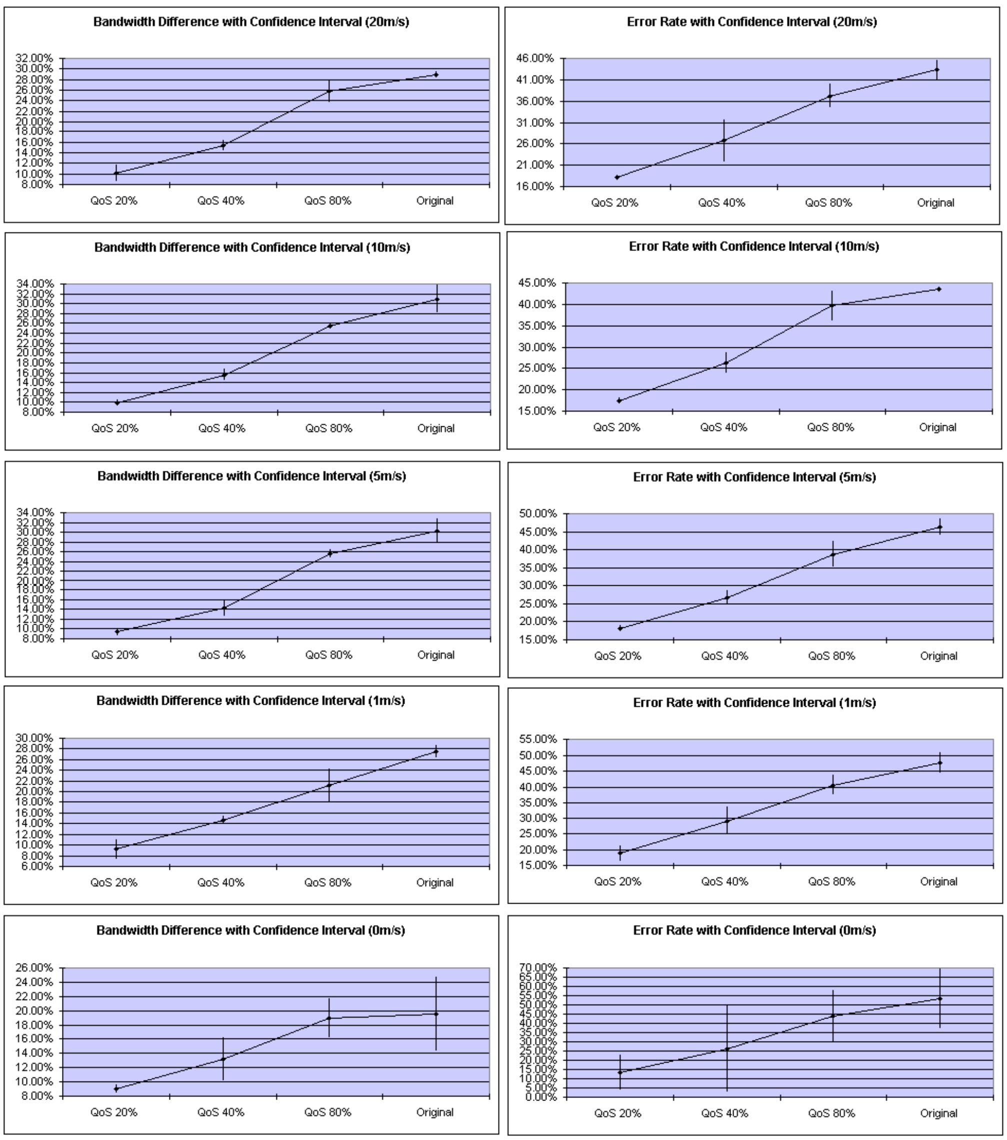


***Figure 21**: QoS Performance Comparison with Confidence Intervals*

From the graphs, it can be seen that for movement patterns 20m/s, 10m/s, 5m/s, and 1m/s, the 20% OLSR's performance improvement over 40% OLSR, 40% OLSR's over 80% OLSR, and 80%'s over the original OLSR in QoS aspect is statistically significant. In the fixed network scenarios, the 4 OLSR algorithms' confidence interval range overlap with one another, except that 20% OLSR's performance improvement over the original OLSR in Bandwidth Difference and Error Rate is statistically significant. This is because in the fixed network, with few TC messages for topology update, the bandwidth conditions on the alternative routes do not significantly differ from each other.

Based on the simulation result presented and analyzed above, it can be seen that the QoS OLSR algorithms do enhance the network QoS performance. However, in order to achieve these improvements, additional "protocol overhead" is also introduced, which degrades the performance of these QoS routing protocols, especially with respect to "Packet Delivery Ratio" and "End-to-End Delay".

As there is a trade-off between the achievements the routing algorithms make and the price that is paid to get such achievement, the routing protocols should be selected carefully based on the request of the data application.

VII. Conclusions

In this report, the authors describe the importance of QoS routing in Ad-Hoc networks, the challenges that were encountered and approaches that were taken.
The authors propose a straightforward method to calculate the available bandwidth over wireless links. Based on the bandwidth calculation algorithm, they discuss in detail how to add support for bandwidth QoS into the OLSR protocol, and introduce three algorithms that allow OLSR to find the maximum bandwidth path. They use a simple network simulator that they have developed to obtain and analyze the simulation results under a number of network snapshots. From a performance perspective, all three algorithms increase the odds of finding a path that is optimal under bandwidth constraints. They also prove that for their network model, two of the algorithms (OLSR_R2 and OLSR_R3) are indeed optimal – find the best bandwidth path.
In addition to analyzing the algorithms based on static network snapshots, the authors also added one of the algorithms proposed (OLSR_R2) to an OLSR simulation using OPNET to explore the impact of node movement and bandwidth change. In the simulations in OPNET, they do not only compare the basic performance and the QoS performance of the original OLSR protocol and the QoS OLSR versions (OLSR_R2 with different parameters), but also analyze the advantages and limitations.

There is a trade-off between the performance the QoS routing protocol achieves and the additional cost it introduces. The QoS OLSR versions that the authors study in this report support the above claim – QoS OLSR algorithms do enhance the network QoS performance. However, in order to achieve this improvement, additional “protocol overhead” is also introduced, which degrades the performance of these QoS routing protocols, especially with respect to “End-to-End Delay” and “Packet Delivery Ratio” in the high speed movement scenarios, where the performance difference is statistically significant. As the added overhead is the main cost that affects the QoS routing algorithm’s performance, the future work on QoS routing in Ad-Hoc networks may be focused on how to reduce the overhead. The following are some basic ideas:

- In the static network simulations, OLSR_R1 does not find the best bandwidth route all the time. However, it has much improvement over the original OLSR protocol, while has almost the same overhead as that of the original OLSR protocol. From the simulations in OPNET, the authors learned that the high overhead is the main reason for the inferior packet delivery ratio performance of the QoS OLSR versions, so it would be interesting to implement OLSR_R1 in OPNET to observe its performance.
- From the analysis of OPNET simulations, the authors observed that the TC packet collisions at the 2-hop neighbors cause the problem of stale routing tables. TC message collisions happen when there are 2 MPRs relaying TC messages at the same time. This problem happens in both the original OLSR protocol and the QoS OLSR versions. To avoid this problem, a jitter mechanism was added into OLSR protocol – when an MPR receives a TC message, it waits for a random delay time before it relays that TC message, instead of relaying it immediately. The random delay in OPNET could be implemented to see if this idea could improve the QoS OLSR packet delivery ratio.
- Compared to the load of data packets, the additional overhead the QoS OLSR versions introduce use a large portion of bandwidth, causing more data packet delay for the QoS OLSR versions. Currently the authors are using 2 Mbps data rate. It would be interesting to explore using 802.11b, with 11 Mbps data rate, and examine whether the added overhead would still have a negative effect with respect to the delay.

- Some of the simulation results (the 0m/s speed scenario in the 50-nodes-network and delay of all scenarios in the 30-nodes-network) have comparatively large confidence intervals. To compare more accurately the performance of the original OLSR protocol and the QoS OLSR versions, more simulations could be run in the future.
- The above future work targets on QoS version of OLSR. However, it would also be interesting to design and implement the pro-active QoS routing based on other best-effort Ad-Hoc network routing protocols and examine the performance. By doing so a more accurate conclusion could be reached as to which kind of QoS routing protocol is more suitable for Ad-Hoc network, link-constrained routing or link-optimization routing.

References

1. G. S. Ahn, A. T. Campbell, A. Veres and L. H. Sun, "SWAN: Service Differentiation in Stateless Wireless Ad-Hoc Networks", IEEE Infocom 2002, pages 457-466, June, 2002
2. G. Apostolopoulos, R. Guerin, S. Kamat, and S. K. Tripathi, "Quality of Service Based Routing: A Performance Perspective", Association for Computing Machinery's Special Interest Group on Data Communication '98, pages 17-28, September 1998
3. S. Chen and K. Nahrstedt, "Distributed Quality-of-Service Routing in High-Speed Networks Based on Selective Probing", IEEE Local Computer Networks (LCN), pages 80-89, October 1998
4. R. Guerin, and Orda. Willimas, "Qos Routing Mechanisms and OSPF Extensions", draft-qos-routing-ospf-00.txt, Internet-Draft, Internet Engineering Task Force, November 1996
5. A. Iwata, C. C. Chiang, G. Pei, M. Gerla and T. Chen, "Scalable Routing Strategies for Ad-Hoc Wireless Networks", IEEE Journal on Selected Areas in Communications, Vol.17, No.8, pages 1369-1379, August 1999
6. P. Jacuet, P. Minet, P. Muhlethaler and N. Rivierre, " Increasing Reliability in Cable-Free Radio LANs Low Level Forwarding in HIPERAN", Wireless Personal Communications, Vol. 4, No. 1, pages 51-63, January 1997
7. P. Jacquet, P. Muhlethaler, A. Qayyum, A. Laouiti, L. Viennot, and T. Clauseen, "Optimized Link State Routing Protocol", draft-ietf-manet-olsr-0.6.txt, Internet Draft, Mobile Ad-Hoc Networking Working Group, March 2002
8. D. B. Johnson and D. A. Maltz, "Dynamic Source Routing in Ad-Hoc Wireless Networks", Mobile Computing, Kluwer Academic Publishers, ISBN: 0792396979, Chapter 5, pages 153-181, 1996
9. C.R.Lin and M.Gerla, "MACA/PR: An Asynchronous Multimedia Multihop Wireless Network", Proceedings of IEEE INFOCOM'97, 1997
10. C. R. Lin and J. S. Liu, "QoS Routing in Ad-Hoc Wireless Networks", IEEE Journal On Selected Areas In Communications, Vol.17, No.8, pages 1426-1438, August 1999
11. J. Macker and S. Corson, IETF Mobile Ad-Hoc Networking Working Group Charter, http://www.ietf.org/html.charters/manet-charter.html
12. Muir and J.J.Garcia-Luna-Aceves, "An Efficient Packet-sensing MAC Protocol for Wireless Networks", ACM Journal on Mobile Networks and Applications, Vol 3, No. 2, pages 221-234, August 1998
13. OPNET, http://www.opnet.com
14. C. E. Perkins, E. M. Royer, and S. R. Das, "Performance Comparison of Two On-Demand Routing Protocols for Ad-Hoc Networks", IEEE Personal Communications, pages 16-28, February 2001
15. A. Qayyum, L. Viennot, and A. Laouiti, "Multipoint Relaying: An Efficient Technique for Flooding in Mobile Wireless Networks", INRIA Research Report RR-3898, February 2000
16. R. Ramanathan and M. Steenstrup, "Hierarchically-Organized, Multihop Mobile Wireless Networks for Quality-of-Service Support", Mobile Networks and Applications, Vol.3, pages 101-119, 1998
17. R. Sivakumar, P. Sinha and V. Bharghavan, "CEDAR: A Core-Extraction Distributed Ad-Hoc Routing Algorithm", IEEE Journal On Selected Areas In Communication, Vol.17, No.8, pages 1454-1465, August 1999
18. J.L.Sobrinho and A.S.Krishnakumar, "Quality-of-Service in Ad-Hoc Carrier Sense Multiple Access Wireless Networks", IEEE Journal on Special Areas in Communications, Vol.17, No.8, August 1999

19. K. Wu and J. Harms, “QoS Support in Mobile Ad-Hoc Networks”, Crossing Boundaries – an Inter Disciplinary Journal, Vol. 1, No. 1, pages 92-107, Fall 2001
20. Z. Zhang and J. Crowcroft, “QoS Routing for Supporting Resource Reservation”, IEEE Journal on Selected Areas in Communications, Vol.14, No.7, pages 1228-1234, September 1996

List of symbols/abbreviations/acronyms/initialisms

DND	Department of National Defence
20% OLSR	Quality of Service version of OLSR with 20% bandwidth updates threshold
40% OLSR	Quality of Service version of OLSR with 40% bandwidth updates threshold
80% OLSR	Quality of Service version of OLSR with 40% bandwidth updates threshold
Ack	Acknowledgement
BF	Bellman-Ford
CEDAR	Core-Extraction Distributed Ad-Hoc Routing
CSMA/CA	Carrier Sense Multiple Access and Collision Avoidance
CTS	Clear To Send
DCF	Distributed Coordination Function
MANET	Mobile Ad-Hoc Network
MPR	Multi-Point Relay
OLSR	Optimized Link State Routing
QoS	Quality of Service
QoS OLSR	Quality of Service versions of OLSR
QoS Routing	Quality of Service Routing
RTS	Request To Send
TC message	Topology Control Message
WLAN	Wireless LAN

DOCUMENT CONTROL DATA

(Security classification of title, body of abstract and indexing annotation must be entered when the overall document is classified)

1. ORIGINATOR (the name and address of the organization preparing the document. Organizations for whom the document was prepared, e.g. Establishment sponsoring a contractor's report, or tasking agency, are entered in section 8.)	2. SECURITY CLASSIFICATION (overall security classification of the document, including special warning terms if applicable)
Communications Research Center Canada 3701 Carling Avenue, PO Box 11490, Station H Ottawa, ON, K2H 8S2	UNCLASSIFIED

3. TITLE (the complete document title as indicated on the title page. Its classification should be indicated by the appropriate abbreviation (S,C or U) in parentheses after the title.)

Proactive QoS Routing in Ad-Hoc Networks (U)

4. AUTHORS (Last name, first name, middle initial)

Ge, Ying, Kunz, Thomas; Lamont , Louise

5. DATE OF PUBLICATION (month and year of publication of document)	6a. NO. OF PAGES (total containing information. Include Annexes, Appendices, etc.)	6b. NO. OF REFS (total cited in document)
April 2003	46	20

7. DESCRIPTIVE NOTES (the category of the document, e.g. technical report, technical note or memorandum. If appropriate, enter the type of report, e.g. interim, progress, summary, annual or final. Give the inclusive dates when a specific reporting period is covered.)

Technical Report DRDC Ottawa TR 2003-075; CRC-RP-2003-005

8. SPONSORING ACTIVITY (the name of the department project office or laboratory sponsoring the research and development. Include the address.)

Defense R&D Canada -- Ottawa
3701 Carling Avenue
Ottawa, ON K1A 0Z4

9a. PROJECT OR GRANT NO. (if appropriate, the applicable research and development project or grant number under which the document was written. Please specify whether project or grant)	9b. CONTRACT NO. (if appropriate, the applicable number under which the document was written)
Work Unit #5cn	

10a. ORIGINATOR'S DOCUMENT NUMBER (the official document number by which the document is identified by the originating activity. This number must be unique to this document.)	10b. OTHER DOCUMENT NOS. (Any other numbers which may be assigned this document either by the originator or by the sponsor)
CRC-RP-2003-005	DRDC Ottawa TR 2003-005

11. DOCUMENT AVAILABILITY (any limitations on further dissemination of the document, other than those imposed by security classification)

(x) Unlimited distribution
(   ) Distribution limited to defence departments and defence contractors; further distribution only as approved
(   ) Distribution limited to defence departments and Canadian defence contractors; further distribution only as approved
(   ) Distribution limited to government departments and agencies; further distribution only as approved
(   ) Distribution limited to defence departments; further distribution only as approved
(   ) Other (please specify):

12. DOCUMENT ANNOUNCEMENT (any limitation to the bibliographic announcement of this document. This will normally correspond to the Document Availability (11). However, where further distribution (beyond the audience specified in 11) is possible, a wider announcement audience may be selected.)

Unlimited

13. ABSTRACT (a brief and factual summary of the document. It may also appear elsewhere in the body of the document itself. It is highly desirable that the abstract of classified documents be unclassified. Each paragraph of the abstract shall begin with an indication of the security classification of the information in the paragraph (unless the document itself is unclassified) represented as (S), (C), or (U). It is not necessary to include here abstracts in both official languages unless the text is bilingual).

(U)Quality-of-service (QoS) routing in mobile Ad-Hoc networks is challenging because the network topology may change constantly and the available state information for routing is inherently imprecise. In this report, the authors have developed different QoS versions of the OLSR (Optimized Link State Routing) protocol, which is a “pro-active” Ad-Hoc routing protocol. They have introduced algorithms that allow OLSR to find the maximum bandwidth path and have shown that these algorithms do improve OLSR in terms of bandwidth. They have also analyzed the performance of the QoS routing protocols in OPNET, observed the results obtained, and the consequences. The simulation results show that the QoS versions of the OLSR routing protocol do improve the available bandwidth of the routes computed, but the added cost – the additional overhead also has a negative impact on the network in End-to-End Delay and Packet Delivery Ratio, especially in the high speed movement scenarios. The authors believe that proactive QoS routing is still worth while studying. Emphasis on future studies should be on how to reduce the overhead of QoS pro-active routing protocols.

14. KEYWORDS, DESCRIPTORS or IDENTIFIERS (technically meaningful terms or short phrases that characterize a document and could be helpful in cataloguing the document. They should be selected so that no security classification is required. Identifiers such as equipment model designation, trade name, military project code name, geographic location may also be included. If possible keywords should be selected from a published thesaurus. e.g. Thesaurus of Engineering and Scientific Terms (TEST) and that thesaurus-identified. If it is not possible to select indexing terms which are Unclassified, the classification of each should be indicated as with the title.)

Ad-Hoc Network
Routing Protocol
Quality of Service
QoS Routing, OLSR

Defence R&D Canada

Canada's leader in defence
and national security R&D

R & D pour la défense Canada

Chef de file au Canada en R & D
pour la défense et la sécurité nationale

DEFENCE R&D DÉFENSE

www.drdc-rddc.gc.ca